YOUR NEXT GENERATION STORE

TRANSFORMACIÓN DIGITAL: Implementación de Estrategias de Ciencia de Datos en DSMarket

ANEXO 2: DESARROLLO MLOPS

Título:

TRANSFORMACIÓN DIGITAL: Implementación de Estrategias de Ciencia de Datos en DSMarket

Capstone Project – Grupo 1 Retail

Autores:

- Juan José Guerrero López
- Adrián Ortega Fernández
- Ariana Puentes Lafée
- Cristina Sánchez García

Tutor:

Daniel Pegalajar Luque

Fecha: 21-marzo-2025

INDICE

1. Introducción

El objetivo de esta propuesta es definir cómo aplicar los modelos predictivos de ventas al abastecimiento de tiendas, establecer un esquema de productivización mediante una API e identificar mejoras en la predicción. Actualmente, DSMarket utiliza métodos tradicionales para estimar la demanda de productos, basándose principalmente en la experiencia. Este enfoque es rudimentario y tiene limitaciones significativas en cuanto a precisión y capacidad para ajustarse rápidamente a cambios en la demanda, lo que resulta en sobreabastecimiento o desabastecimiento de productos.

El modelo de Machine Learning (ML) propuesto busca reemplazar estos métodos con predicciones basadas en datos históricos, tendencias, estacionalidad, etc, lo que permite una gestión más precisa y dinámica del inventario. Este modelo no solo optimiza las decisiones de abastecimiento, sino que también mejora la eficiencia operativa minimizando los costos asociados con el exceso de stock y la falta de productos. Nuestro modelo nos ayudará a evitar pérdidas y desabastecimiento.

Para gestionar de manera eficiente la implementación y mantenimiento de este modelo, se empleará un sistema de MLOps, que automatiza los flujos de trabajo de desarrollo, despliegue y monitorización del modelo. Este sistema MLOps se apoya en Google Cloud Platform (GCP) y herramientas como Cloud Run, FastAPI y AlloyDB, lo que garantiza que las predicciones y recomendaciones de reabastecimiento se realicen de manera ágil, escalable y continua, sin intervención manual.

2. Aplicación del modelo de predicción al abastecimiento de tiendas

La integración del modelo de predicción con el proceso de abastecimiento actual de DSMarket nos permite mejorar la eficiencia del inventario, reducir las pérdidas y minimizar los stock-outs o desabastecimientos, lo que mejora la experiencia del cliente y optimiza los márgenes operativos.

2.1. Objetivo del modelo de predicción

El objetivo del modelo es predecir la demanda futura de productos en cada tienda de DSMarket. Se realizan predicciones teniendo en cuenta la variabilidad de la demanda, que junto con los datos logísticos y de almacenamiento nos permiten determinar las cantidades óptimas de abastecimiento. La previsión se realiza sobre una ventana de 28 días (cuatro semanas), lo que proporciona suficiente tiempo para planificar los pedidos de reposición.

2.2. Integración con el proceso de abastecimiento

Actualmente, el modelo de predicción de ventas se integra con datos históricos provenientes de las tiendas que han sido los datos usados para entrenar el modelo.

Para completar el proceso de abastecimiento es necesario integrarlo con datos logísticos y de inventario que actualmente no están siendo considerados en el modelo. Es necesario comprobar si estos datos están a nivel de producto o de tienda para llevar a cabo un cálculo de stock óptimo.

En próximos pasos se puede implementar un nuevo modelo de Machine Learning para procesar datos logísticos y ayudar a predecir el stock óptimo a nivel producto-tienda.

2.3. Esquema del Flujo de Datos

A continuación, se detalla cómo los datos fluyen desde la entrada hasta las recomendaciones finales de reabastecimiento.

2.3.1. Entrada de Datos:

- Datos históricos de ventas: Se utilizan los registros de ventas de cada tienda para analizar patrones y comportamientos de la demanda de productos.
- Calendario de eventos: Datos sobre festividades, promociones, y otros eventos que pueden afectar la demanda de productos.
- Precios: Cambios de precios y descuentos aplicados a productos que pueden modificar el comportamiento de compra de los clientes.

2.3.2. Predicción:

- Usando los datos de entrada, el modelo genera predicciones de la demanda futura para cada producto, por tienda, para las siguientes cuatro semanas.

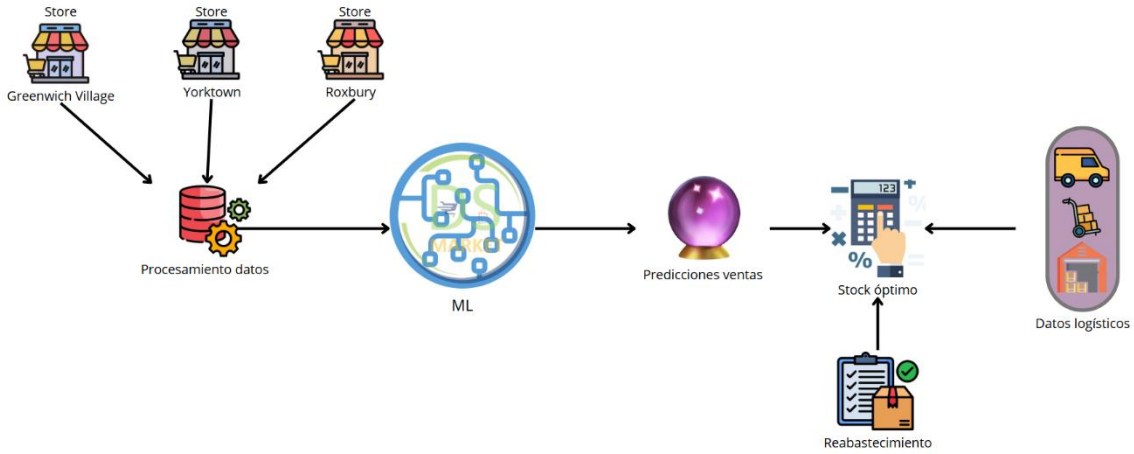
2.3.3. Optimización:

Se incorporan los datos logísticos y se calcula el stock óptimo para cada tienda, considerando factores como:

- Tiempo de entrega: Cuánto tiempo toma recibir los productos desde el almacén.
- Capacidad de almacenamiento: El espacio disponible en cada tienda para mantener los productos.
- Nivel de stock: Determinado para evitar tanto el desabastecimiento como el exceso de inventario.

2.3.4. Recomendaciones de reabastecimiento:

Con todos estos datos somos capaces de crear recomendaciones de cuánto pedir de cada producto en cada tienda para satisfacer la demanda proyectada. Estas recomendaciones se pueden integrar con las plataformas internas de gestión de pedidos para una ejecución eficiente.



2.4. Beneficios esperados

Al aplicar este modelo de predicción al abastecimiento de tiendas, DSMarket podrá:

- Optimizar la gestión del inventario: Se pueden realizar pedidos más ajustados a las necesidades reales de las tiendas, reduciendo tanto el exceso de inventario como los productos agotados.
- Reducir pérdidas por sobreabastecimiento: Evitar tener productos en exceso, lo que puede llevar a pérdidas económicas por deterioro de productos o almacenamiento innecesario.
- Minimizar desabastecimientos: Garantizar que las tiendas tengan suficiente stock de los productos más demandados, evitando la falta de productos que afecten las ventas y la satisfacción del cliente.
- Mejor planificación logística: Con las predicciones de demanda y las recomendaciones de reabastecimiento, se optimiza la logística, lo que puede reducir costos de transporte y mejorar la eficiencia en la distribución de productos.

2.5. Productivización del modelo

En este punto explicamos cómo hemos pasado de un modelo experimental de predicción de ventas a una solución operativa que se puede usar de manera continua dentro de la infraestructura de la empresa.

Para integrar la solución dentro de la operación de DSMarket, se ha desplegado una API en Google Cloud Run. Esta API permitirá que los sistemas internos consulten predicciones en tiempo real.

2.6. Arquitectura de la Solución

La solución se despliega en Google Cloud Platform (GCP) utilizando una serie de herramientas que permiten escalabilidad automática, alta disponibilidad y portabilidad.

2.6.1. Despliegue en la nube:

La API de predicción se desplegará en Google Cloud Run, lo que permite que el sistema escale automáticamente según la demanda de tráfico, sin necesidad de intervención manual.

FastAPI se utiliza para exponer el modelo como una API accesible para hacer predicciones en tiempo real.

2.6.2. Contenerización con Docker:

Usamos Docker para contenerizar el modelo y sus dependencias. Esto garantiza que el entorno de desarrollo y el de producción sean consistentes, facilitando el proceso de despliegue y manteniendo la infraestructura limpia.

Dockerfile utilizado para crear la imagen:

```
FROM python:3.9
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
CMD ["uvicorn", "api:app", "--host", "0.0.0.0", "--port", "8080"]
```

2.6.3. Base de datos AlloyDB en Google Cloud

Se eligió AlloyDB en Google Cloud frente a otras opciones por su alto rendimiento para cargas de trabajo, como las predicciones de ventas y la consulta de datos históricos, es hasta 4 veces más rápido que PostgreSQL tradicional.

Capacidad de procesar transacciones y análisis en paralelo (HTAP), se pueden realizar consultas analíticas y realizar predicciones en tiempo real.

En esta base de datos en la nube almacenamos los datos históricos de ventas y los resultados de las predicciones.

2.6.4. Despliegue del modelo en Google Cloud Run

Actualmente el modelo se entrena y almacena en local para posteriormente crear una imagen Docker que lanzaremos en Google Cloud Run.

La escalabilidad es automática, se ajusta en función de la demanda y además nos ahorramos gestionar servidores o Kubernetes. Se paga sólo por los recursos utilizados.

Se ha desplegado en la región europe-west1 por ser la más cercana a Madrid.

2.6.5. Gestión del modelo de predicción de ventas.

Por motivos de tiempo para esta primera prueba no se ha implementado un sistema para gestionar el modelo de predicción. Como hemos explicado el modelo se entrena localmente y cuando termina se crea la imagen Docker para Google Cloud Run.

El siguiente paso a implementar es usar MLFlow para gestionar el ciclo de vida del modelo. De esta manera podremos entrenar el modelo con los datos históricos almacenados en AlloyDB.

MLFlow permite realizar un seguimiento de los experimentos, gestionar diferentes versiones de los modelos entrenados y el registro de resultados.

Este modelo se conteniza en una imagen Docker y se despliega automáticamente en Google Cloud run siendo el flujo a partir de aquí idéntico al actual.

Para una futura versión, también haríamos uso de Google Cloud Logging para la revisión y análisis del modelo.

2.6.6. Frontend con Github Pages

Se ha creado una interfaz web para esta primera versión. Esta página web sirve para hacer pruebas siendo estática y accesible para todo el mundo.

Incluye un formulario interactivo que permite al usuario ingresar el producto y tienda que desea predecir. La fecha para la predicción por defecto es el último día que tenemos en los datos no pudiéndose elegir otra.

```
<script>
    async function hacerPrediccion() {
        const datos = {/* datos de entrada */};
        const respuesta = await fetch("https://sales-api-xyz123-ew.a.run.app/predict", {
            method: "POST",
            headers: {"Content-Type": "application/json"},
            body: JSON.stringify(datos)
        });
        const resultado = await respuesta.json();
        document.getElementById("resultado").innerText = "Predicción de Ventas: " + resultado.predicted_sales;
    }
</script>
```

2.6.7. CI/CD con Google Cloud Build y GitHub Actions

Se ha creado una organización en GitHub en la que se encuentran los repositorios con todo el código necesario para hacer funcionar la API y la web. Puedes acceder a él pulsando aquí.

El ciclo de vida del modelo y el proceso de despliegue se automatizan con **Google Cloud Build** y **GitHub Actions**, lo que asegura que cada vez que se actualicen los datos o el código, se reconstruya la imagen Docker y se despliegue automáticamente en Cloud Run.

Google Cloud Build se configura mediante un archivo *cloudbuild.yaml* que automatiza la construcción y el despliegue.

```
steps:
  # imagen Docker
  - name: 'gcr.io/cloud-builders/docker'
    args: ['build', '-t', 'europe-west1-docker.pkg.dev/mlops-nuclio-022025/ds-market/sales-api', '.']

  # imagen a Artifact Registry
  - name: 'gcr.io/cloud-builders/docker'
    args: ['push', 'europe-west1-docker.pkg.dev/mlops-nuclio-022025/ds-market/sales-api']

  # imagen a Cloud Run
  - name: 'gcr.io/cloud-builders/gcloud'
    args: ['run', 'deploy', 'sales-api', '--image', 'europe-west1-docker.pkg.dev/mlops-nuclio-022025/ds-market/sales-api', '--platform', 'managed', '--region', 'europe-west1', '--allow-unauthenticated']
```

De esta manera cuando hago *push* en el repositorio **dsmarket-api** el activador *deploy-cloud-run* de Cloud Build construye y despliega automáticamente la API en Cloud Run.

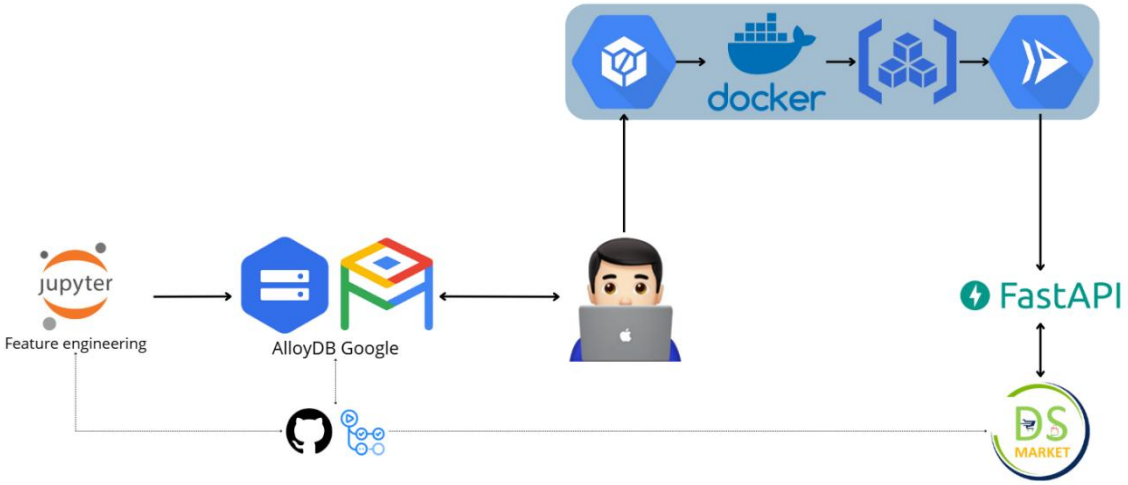
Por otro lado, cuando hago un *push* en **dsmarket-web**, GitHub Actions lanza el workflow *deploy-from-a-branch* actualiza la web automáticamente en **GitHub Pages**.

2.7. Endpoints de la API

1️⃣ */predict* → Devuelve la predicción de ventas para un producto/tienda en los 28 días posteriores.

Esto permitirá que el equipo de Operaciones acceda a recomendaciones optimizadas de reabastecimiento en tiempo real.

Con el objetivo de mejorar la precisión del modelo y adaptarlo mejor a las necesidades del abastecimiento, se proponen dos líneas de mejora complementarias.



En primer lugar, se abordarán las relacionadas con los datos, incluyendo su obtención, procesamiento y calidad. A continuación, se presentarán las extensiones directamente vinculadas al modelo en sí, como su arquitectura, entrenamiento y ajuste.

3. Preprocesamiento y enriquecimiento de datos

Se proponen las siguientes mejoras con el objetivo de mejorar la calidad y cantidad de los datos.

3.1. Incorporación de datos logísticos

Para mejorar el sistema de abastecimiento se deben considerar diversos factores logísticos:

3.1.1. Tiempo de entrega

Los tiempos de entrega entre los almacenes y las tiendas afectan la cantidad de stock que debe pedirse con antelación. Los modelos deben ser capaces de estimar la demanda anticipada, teniendo en cuenta el tiempo de entrega de los productos.

3.1.2. Disponibilidad en almacenes

Es fundamental tener en cuenta los niveles de inventario en los almacenes centrales. Si un producto está agotado en el almacén, no se podrá hacer la reposición de manera inmediata, lo que puede afectar la precisión de la predicción de la demanda.

3.1.3. Capacidad de almacenamiento de las tiendas

Algunas tiendas tienen un espacio limitado para almacenar productos, lo que significa que deben ajustarse a un límite de stock. El modelo debe prever estos límites y ajustar las recomendaciones de reabastecimiento para no exceder la capacidad de cada tienda.

3.2. Segmentación avanzada de productos y departamentos

Mejorar la información sobre las categorías y departamentos de los productos mejora la segmentación de la demanda y permite ajustar mejor las estrategias de abastecimiento.

Sería interesante incluir información sobre la caducidad de los productos.

- Productos frescos: La demanda de estos productos varía mucho dependiendo de la estacionalidad y los cambios en el clima. El modelo debe poder ajustarse para predecir la demanda a corto plazo, especialmente para minimizar el riesgo de caducidad.
- Electrodomésticos y artículos no perecederos: La demanda puede depender más de promociones o tendencias de mercado, lo que hace que el modelo se ajuste a estas fluctuaciones con menos regularidad que los productos frescos.

Estas mejoras permitirán que el sistema no solo prediga ventas, sino que optimice la decisión de cuánto stock reponer en cada tienda.

1.1. Incorporación de factores internos

Para obtener predicciones más precisas y alineadas con la demanda real del mercado, es importante incorporar factores externos que afecten las ventas:

3.2.1. Eventos especiales

En el análisis efectuado se ha demostrado el impacto que tienen los eventos en el número de ventas por eso consideramos que el modelo debe estar preparado para incluir información más detallada sobre estos datos.

3.2.2. Precios y promociones

La dinámica de precios tiene un impacto directo sobre la demanda de productos. Actualmente existen datos de precios pero no se registran si los cambios que sufren los precios se deben a promociones o rebajas. Incluir esto a nivel global o regional puede ser muy beneficioso.

3.3. Incorporación de factores externos

Los cambios en el entorno social, económico y cultural pueden tener un impacto significativo sobre el comportamiento de los consumidores y, por lo tanto, sobre la demanda de productos.

3.3.1. Festivos nacionales y locales

Incluir un calendario que incluya estos festivos y eventos importantes. Esto es algo que ya hemos implementado en nuestro modelo actual con un calendario estándar pero quizás la empresa tiene un calendario propio que se ajusta mejor a nuestras necesidades.

3.3.2. Datos demográficos

Las características demográficas de la población en las áreas geográficas donde operan las tiendas pueden influir en la demanda. Por ejemplo, en áreas con una mayor concentración de familias jóvenes, podría haber una mayor demanda de productos infantiles.

El modelo puede incorporar variables demográficas como edad promedio, nivel de ingresos, tamaño familiar, etc.

3.3.3. Factores económicos

Los factores económicos, como la inflación, tasas de desempleo, o cambios en el poder adquisitivo de los consumidores, pueden afectar a la demanda. Aunque no sea de vital importancia en el caso de uso de reabastecimiento, es interesante incluir este tipo de datos para análisis a largo plazo y detección de anomalías.

El modelo debe ser capaz de integrar indicadores económicos de fuentes externas (por ejemplo, índices económicos nacionales o locales).

3.3.4. Condiciones climáticas

Las condiciones climáticas (temporada de lluvias, huracanes) pueden influir en la demanda de ciertos productos, especialmente aquellos relacionados con el clima y más cuando estos sucesos climáticos suceden con cierta estacionalidad.

1.2. Automatización de cálculo de stock óptimo:

Definir umbrales de reabastecimiento según las predicciones.

3.4. Definir umbrales de stock mínimo y máximo

Estos umbrales son calculados en función de las predicciones de demanda realizadas por el modelo, los tiempos de entrega y la capacidad de almacenamiento de cada tienda.

- Stock mínimo: La cantidad mínima de un producto que debe estar disponible para evitar desabastecimientos.
- Stock máximo: El límite de unidades de un producto que se debe almacenar para evitar el exceso de inventario, lo que puede resultar en pérdidas por productos no vendidos o vencidos.

3.5. Cálculo Automático de las cantidades de reabastecimiento:

Una vez definidos los umbrales, el sistema puede calcular automáticamente las cantidades necesarias para reponer el stock en función de la diferencia entre el stock actual y el stock mínimo predicho por el modelo.

3.5.1. Ajuste en tiempo real:

A medida que cambian las predicciones de demanda debido a cambios en eventos, promociones o estacionalidad, los umbrales de reabastecimiento se ajustan automáticamente. Esto asegura que el modelo esté siempre optimizado para las condiciones cambiantes del mercado.

3.5.2. Predicción de stock-outs:

Identificación de productos con alto riesgo de agotamiento.

El modelo de predicción de ventas también debe ser capaz de detectar posibles stock-outs (agotar existencias) y optimizar las decisiones de reabastecimiento:

3.5.3. Detección de rotura de stock:

A través de modelos de clasificación, el sistema puede identificar productos que tienen un alto riesgo de agotarse. Este modelo se entrena utilizando datos históricos de ventas y el comportamiento de la demanda, lo que permite predecir qué productos estarán en riesgo de no estar disponibles en el futuro cercano.

3.5.4. Optimización del reabastecimiento:

Basado en la predicción de demanda y en la disponibilidad de inventarios, el sistema puede sugerir las cantidades óptimas de reposición. Este proceso tiene en cuenta los tiempos de entrega, la capacidad de almacenamiento y las limitaciones de los almacenes, buscando minimizar los costos de sobreabastecimiento y desabastecimiento.

4. Extensiones en la arquitectura y entrenamiento del modelo

A medida que el sistema de predicción de ventas evoluciona, se han identificado varias líneas de mejora orientadas a aumentar la precisión, escalabilidad y flexibilidad de la solución actual. Estas extensiones permiten anticipar nuevas necesidades del negocio y garantizar una integración más fluida dentro de la infraestructura tecnológica de DSMarket.

4.1. Modelos más profundos y computación escalable

Con más tiempo de entrenamiento y acceso a mayores recursos computacionales (Cloud Run + AlloyDB), se abre la posibilidad de emplear modelos más complejos y profundos que puedan captar mejor las interacciones no lineales entre variables, especialmente en contextos de alta estacionalidad o con patrones de venta muy específicos.

4.2. Mejora en la asignación de pesos a productos

Actualmente, el modelo middle-to-bottom calcula las ventas a nivel de producto (id) distribuyendo las predicciones del clúster a los productos individuales según su histórico reciente (30 días). Esta aproximación puede resultar limitada si no se consideran dinámicas del mercado.

Esto se podría mejorar implementando un modelo de regresión auxiliar que calcule estos factores de asignación utilizando variables como:

- Media de precios y variación
- Tendencias recientes de ventas
- Fechas y eventos
- Información regional
- Categoría y estacionalidad del producto

Esto permitirá ajustar los pesos dinámicamente según el contexto actual y no solo en base al histórico.

4.3. Mejoras en el sistema MLOps

Aunque el sistema MLOps ya se encuentra operativo, se han identificado varias áreas de expansión:

- Migración completa a la nube: se procederá a subir todo el histórico a AlloyDB, incluyendo actualizaciones en tiempo real mediante pipelines automatizados.
- Ejecución del modelo online: se desplegará una versión del modelo entrenado con seguimiento a través de MLFlow.
- Automatización y mantenimiento: se reforzará el pipeline de CI/CD en Google Cloud Build, y se habilitará el logging centralizado mediante Google Cloud Logging.

Estas medidas permitirán una mayor trazabilidad y fiabilidad del sistema en producción.

4.4. Perfiles personalizados en la API

Se plantea la incorporación de una capa de autenticación basada en perfiles de usuario dentro de la API:

- Administrador: acceso total, posibilidad de cambiar fecha de predicción y parámetros del modelo.
- Gestor de tienda: acceso restringido a su tienda y fechas específicas.
- Analista de datos: acceso a logs, históricos y predicciones por lote.

Esto permitirá una personalización del acceso según los distintos roles dentro de DSMarket, alineándose con las necesidades operativas y de seguridad.

5. Diseño de prueba piloto

Antes de implementar la aplicación de modelos de predicción en DS Market a gran escala, se realizará un piloto para evaluar su efectividad.

Con esta prueba queremos demostrar el impacto económico positivo que tendrá en la empresa el uso de modelos de predicción para el caso de uso de reabastecimiento de productos.

5.1. Descripción enfoques

A continuación se definen los dos enfoques actuales:

- Rudimentario: media móvil(***). Enfoque basado en reglas simples y procesos manuales para prever ventas e inventarios, con márgenes de error altos que afectan a la toma de decisiones.

- Machine Learning: modelo de predicción de ventas basado en datos históricos de las tiendas de New York, Boston y Philadelphia.

5.2. Metodología de la prueba piloto

Selección de KPIs
Ventas por producto, margen de ganancia, niveles de inventarios, precisión de las predicciones, costos operativos, y eficiencia en la reposición de inventarios.
Selección de productos y tiendas
Se probarán productos en las 10 tiendas distribuidas entre Nueva York, Boston y Filadelfia. Las categorías y departamentos serán utilizados para segmentar los resultados y analizar los beneficios de ML en distintos segmentos de productos.
Ejecutar ambos enfoques
Se ejecutarán ambos enfoques (tradicional y ML) durante un período determinado (por ejemplo, 12 semanas), y se recolectarán los datos necesarios para la comparación.

5.3. Resultados y análisis comparativo

Durante las 12 semanas de prueba, se recopilarán datos sobre ventas, márgenes de ganancia y predicciones de ventas para los productos de las 10 tiendas.

Posteriormente, se llevará a cabo un análisis estadístico utilizando métricas como t-test, RMSE y R2 para comparar la precisión de los enfoques tradicional y de Machine Learning, con el fin de evaluar la efectividad del modelo en comparación con el método anterior.

5.4. Evaluación de la mejora y ROI

Se da por hecho que después de analizar los resultados y comparar ambos enfoques el modelo de predicción de ventas obtendrá un mejor resultado que el enfoque antiguo.

Por lo tanto, una vez conocemos cómo mejora el sistema de Machine Learning al sistema previo, podemos evaluar cómo va a afectar a la empresa implementar este nuevo sistema a nivel global.

Se calculará el retorno de inversión (ROI) comparando los costos de implementación y operación del modelo de ML con los beneficios tangibles obtenidos, que incluyen la optimización de procesos, la mejora de márgenes y la capacidad de toma de decisiones más precisas.

5.5. Presentación de resultados finales

Se presentará un informe ejecutivo con gráficos y tablas comparativas para mostrar los resultados de ambos enfoques en función de los KPIs clave, facilitando una comparación directa entre el enfoque tradicional y el de Machine Learning.

5.6. Próximos pasos

Si los resultados son positivos, se recomendará la adopción de ML a gran escala para todas las tiendas de DSMarket.

Se propondrá un plan de escalabilidad explicando cómo llevar el modelo de predicción al resto de la compañía con un en enfoque en la mejora continua del modelo llevando a cabo la implementación de las extensiones y mejoras que se han expuesto a lo largo de esta propuesta como pueden ser:

- Añadir nuevos datos al modelo:
  - Logísticos
  - Climatológicos
  - Demográficos
  - Competencia
- Implementación modelo de stock óptimo
- Transferencia definitiva a la nube y API mejorada

6. Conclusión

Con esta estrategia, DSMarket podrá aprovechar la inteligencia de datos para optimizar su abastecimiento de tiendas, reducir pérdidas por exceso de inventario y mejorar la disponibilidad de productos. La combinación de un modelo mejorado con una API escalable permitirá que las decisiones de reabastecimiento sean más precisas y automatizadas.

Con esta estrategia, DSMarket no solo podrá optimizar su abastecimiento de tiendas, sino que también se posicionará para reducir las pérdidas por exceso de inventario y mejorar significativamente la disponibilidad de productos.

Hemos logrado crear una solución MLOps simple pero totalmente funcional, capaz de integrar de manera fluida el modelo de predicción de ventas con una API escalable y automatizada. Esta integración permite que las decisiones de reabastecimiento se tomen de forma más precisa y eficiente, apoyadas por un sistema de CI/CD robusto, asegurando que el modelo y los datos estén siempre actualizados, sin necesidad de intervención manual.

Esta solución no solo es escalable, sino que también garantiza rápidos ciclos de entrenamiento, validación y despliegue, permitiendo a DSMarket adaptarse rápidamente a las necesidades del negocio.