Nombre: Ángel Fernández Galán
Dni: 08896350L

Se pide

Contestar justificando todas y cada uno de los apartados y las decisiones tomadas:

1 - Describir la forma de productivizar cada modelo. ¿Qué tipo de herramienta se debería elegir para desarrollar cada modelo y por qué se ha elegido? ¿Cuál sería la forma de acceder a los modelos por parte de los usuarios? ¿Cada cuánto tendrían que reentrenarse?

2 - Describir el flujo de trabajo MLOps de los modelos. ¿Sería necesario en todos los modelos?

3 - ¿Qué cambios habría que hacer en los apartados anteriores si la empresa, en vez de 15 trabajadores, tuviese 10000?

4 - Productivizar el modelo de *bank marketing*. Hay dos opciones para elegir: mediante el diseñador de Azure o bien de forma manual en su propia máquina (serializando los modelos/transformaciones con Pickle).

a. Si el objetivo del modelo es lograr predecir previo a que se realice la llamada qué personas tienen una alta probabilidad de contratar, ¿se podría decir si hay alguna variable que no sea posible incluir en el modelo en producción?, ¿Cuáles son?
b. Probar varios modelos e ingeniería de características y dar con el modelo que mejor métrica presente.

Una vez elegido el mejor modelo, crear una llamada API para consultar el modelo. En caso de haber elegido hacer un modelo de forma manual, se tendrá que generar el código para responder a la llamada *GET* o *POST* con los nuevos datos. En caso de haber elegido hacerlo con Azure, se deberá crear el *endpoint* (hay que recordar que, posiblemente, al usar una cuenta gratuita, no se pueda ejecutar debido a que necesitaría más CPU de las que está permitido usar; no obstante, se deberá plantear la canalización y adjuntar captura de pantalla de ella aunque dé error).

1 - Describir la forma de productivizar cada modelo. ¿Qué tipo de herramienta se debería elegir para desarrollar cada modelo y por qué se ha elegido? ¿Cuál sería la forma de acceder a los modelos por parte de los usuarios? ¿Cada cuánto tendrían que reentrenarse?

Para productivizar los modelos de ML y permitir que los usuarios accedan a ellos, se puede considerar el siguiente enfoque para cada modelo en el escenario dado:

- Conversión de base de datos a CSV:

Para las funciones que extraen datos de las bases de datos y los convierten a archivos CSV en un almacenamiento distribuido, se recomienda utilizar Microsoft Azure Functions. Este servicio permite implementar funciones serverless que se ejecuten de manera programada, como a las 2:00 a. m., y que guarden los archivos generados en Azure Blob Storage o Azure Data Lake Storage**.** Azure Functions es una arquitectura basada en eventos, ideal para procesos automatizados y periódicos como la extracción y almacenamiento de datos en CSV.

Nombre: Ángel Fernández Galán
Dni: 08896350L

• Modelo de predicción de ventas:

Para desarrollar el algoritmo de ML para predecir las ventas futuras existen varias opciones según la complejidad y los requisitos específicos del modelo:

1. **Desarrollo del modelo**

Se pueden utilizar bibliotecas como Scikit-learn, TensorFlow o PyTorch para implementar algoritmos personalizados de predicción basados en técnicas de regresión.

También se puede optar por Azure Machine Learning (Azure ML), que proporciona:

- Integración con herramientas populares de ML.
- Infraestructura escalable para entrenamiento.
- Capacidades de AutoML para experimentación automatizada.

2. **Implementación**

Azure ML permite exponer el modelo como un endpoint web o servicio contenerizado mediante Azure Kubernetes Service (AKS).

Los usuarios pueden acceder al modelo a traves de una API REST, lo que facilita la integracion con sistemas externos como herramientas de BI o sistemas ERP.

3. **Frecuencia de reentrenamiento:**

Este modelo debería reentrenarse periódicamente para reflejar cambios en las tendencias de ventas. La periodicidad puede variar entre semanal o mensual, dependiendo de la estabilidad de los patrones históricos y la disponibilidad de datos actualizados.

• Modelo de predicción de compras telefónicas:

1. **Desarrollo del modelo:**

Nombre: Ángel Fernández Galán
Dni: 08896350L

Similar al modelo de predicción de ventas, se pueden utilizar frameworks como Scikit-learn o TensorFlow para desarrollar el modelo.

Alternativamente, Azure ML ofrece experimentación, entrenamiento y evaluación automatizados para encontrar el mejor algoritmo de clasificación binaria.

2. **Implementación:**

   La implementación se realiza de forma similar, exponiendo el modelo como un endpoint a través de Azure ML, que los usuarios pueden consultar mediante llamadas GET o POST.

3. **Frecuencia de reentrenamiento:**

   Este modelo puede necesitar actualizaciones más frecuentes si el comportamiento de los clientes cambia con rapidez, idealmente cada 2-4 semanas, dependiendo de la cantidad de datos nuevos disponibles.

• Algoritmos de análisis empresarial:

Para los algoritmos que se entrenan y ejecutan en puntos específicos para determinar aspectos determinados del negocio, la elección de las herramientas depende de la naturaleza y complejidad de los algoritmos. Algunos factores a considerar son el tipo de análisis requerido, el tamaño y complejidad de los datos y los algoritmos específicos que se utilizan.

1. **Desarrollo:**

   Se pueden utilizar bibliotecas como Pandas, NumPy y Scikit-learn para análisis exploratorio y algoritmos estadísticos.

   Para conjuntos de datos grandes o análisis complejos, servicios como Azure Databricks o Azure Synapse Analytics permiten realizar análisis distribuidos y procesamiento masivo.

2. **Frecuencia de ejecución:**

   Estos algoritmos pueden ejecutarse de forma puntual o programada, según las necesidades del negocio. En casos específicos, el reentrenamiento o ejecución puede depender de eventos (nuevos datos) o necesidades ad hoc

Nombre: Ángel Fernández Galán
Dni: 08896350L

4. **Acceso de los usuarios a los modelos**

   Los modelos deben exponerse como APIs REST para facilitar la interacción: Las aplicaciones del negocio, como sistemas ERP o herramientas de BI (Power BI, Celonis), pueden enviar solicitudes para obtener predicciones.

   Las APIs pueden integrarse con interfaces personalizadas para los usuarios finales.
   Azure ML ofrece soporte integrado para la creación de endpoints escalables y seguros, lo que permite gestionar el acceso y el control de las predicciones desde cualquier sistema conectado.

2 *- Describir el flujo de trabajo MLOps de los modelos. ¿Sería necesario en todos los modelos?*

El flujo de trabajo de MLOps está diseñado para operacionalizar y gestionar modelos de Machine Learning de manera eficiente a lo largo de su ciclo de vida. Aunque no todos los modelos requieren la misma complejidad en su implementación, aplicar prácticas de MLOps asegura una integración fluida en los procesos empresariales, además de garantizar mantenibilidad, escalabilidad y monitoreo constante. A continuación, se detallan las estrategias recomendadas para cada componente del caso práctico:

1. **Conversión de bases de datos a CSV**

   Esta tarea implica funciones serverless que se ejecutan periódicamente para transformar y almacenar datos en formato CSV. Aunque no es estrictamente una tarea de Machine Learning, ciertas prácticas básicas de MLOps pueden mejorar su fiabilidad:

   - Control de versiones del código: Permite rastrear y gestionar cambios en las funciones serverless (por ejemplo, utilizando Git o Azure Repos).
   - Monitoreo y registro: Implementar logs y alertas para identificar errores en las ejecuciones diarias o retrasos en el procesamiento.

   Con respecto a la pregunta de: ¿Es necesario aplicar MLOps?

   - No es indispensable un flujo completo de MLOps, ya que no hay modelos de ML involucrados. Sin embargo, asegurar la calidad del código y registrar la ejecución mejora la estabilidad del sistema.

Nombre: Ángel Fernández Galán
Dni: 08896350L

2. Modelo de predicción de ventas

Este modelo, utilizado para prever las necesidades de stock, tiene un impacto directo en la toma de decisiones estratégicas. Por ello, un flujo completo de MLOps es altamente recomendado:

- **Recopilación y preparación de datos:** Automatizar pipelines para limpiar y estructurar datos históricos de ventas.
- **Entrenamiento y validación del modelo:** Integrar un pipeline en Azure ML que permita probar distintos algoritmos y ajustar hiperparámetros automáticamente.
- **Implementación:** Publicar el modelo como un servicio API accesible desde sistemas ERP o herramientas de BI.
- **Monitoreo:** Supervisar el desempeño del modelo en producción mediante métricas como precisión y errores de predicción.
- Reentrenamiento programado: Configurar triggers para reentrenar el modelo cuando cambien las tendencias de ventas o haya datos nuevos.

En cuanto a la pregunta de: ¿Es necesario aplicar MLOps?

- Sí, debido a la naturaleza crítica de este modelo, MLOps garantiza que el sistema sea confiable, adaptable y fácil de mantener.

3. Modelo de predicción de compras telefónicas

Al igual que el modelo de predicción de ventas, este sistema requiere una implementación robusta, ya que afecta directamente las estrategias de marketing de la empresa. Las prácticas de MLOps recomendadas son:

- **Automatización del pipeline de datos:** Garantizar que los datos de clientes se procesen y almacenen adecuadamente.
- **Validación continua:** Establecer métricas específicas como recall y F1-score para asegurar que las predicciones sean precisas, especialmente en clases minoritarias (clientes que responden "sí").
- **Adaptación al comportamiento de los clientes:** Diseñar un pipeline de reentrenamiento para reflejar cambios en los patrones de compra.

Con respecto a la pregunt de: ¿Es necesario aplicar MLOps?

Nombre: Ángel Fernández Galán
Dni: 08896350L

- Sí, para mantener la precisión del modelo y ajustar las estrategias de marketing a tiempo, se requiere un flujo de MLOps bien definido.

4. Algoritmos de análisis empresarial

Estos algoritmos suelen ejecutarse de manera puntual para responder a preguntas específicas del negocio. Aunque no necesitan un flujo de MLOps tan sofisticado, algunas prácticas son útiles:

- **Control de versiones y reproducibilidad:** Mantener registros claros del código y los datos utilizados en cada análisis.
- **Pipelines simples para datos:** Automatizar el procesamiento de datos para evitar errores manuales.
- **Pruebas y monitoreo básico:** Verificar la integridad de los datos y resultados para garantizar análisis confiables.

Con respecto a la pregunt de: ¿Es necesario aplicar MLOps?

- No es imprescindible un flujo completo de MLOps, pero aplicar principios básicos mejora la calidad y confiabilidad de los análisis.

3 - ¿Qué cambios habría que hacer en los apartados anteriores si la empresa, en vez de 15 trabajadores, tuviese 10000?

Si la empresa tuviera 1000 trabajadores en lugar de 15, habría ciertos cambios y consideraciones que se deberían implementar en las respuestas anteriores. Veamos las modificaciones requeridas para cada aspecto:

1. Estrategia de análisis en la nube

La conversión de bases de datos a CSV seguiría siendo esencial, pero el sistema tendría que manejar un mayor volumen de datos y usuarios concurrentes:

- **Escalabilidad del almacenamiento:** Utilizar Azure Data Lake Storage Gen2 para soportar grandes volúmenes de datos y accesos simultáneos.
- **Ejecución distribuida:** Las funciones serverless podrían complementarse con pipelines de procesamiento de datos distribuidos en Databricks o Synapse Analytics para manejar picos de demanda.

- **Optimización de costos:** Implementar políticas de gobernanza de datos para evitar un crecimiento descontrolado del almacenamiento y uso de recursos.

2. Desarrollo de modelos de ML

Con una plantilla más numerosa, el desarrollo de modelos debe ser más eficiente y escalable:

- **Computación distribuida:** Utilizar herramientas como Databricks ML para entrenar modelos en paralelo con datasets más grandes.
- **AutoML avanzado:** Implementar Azure AutoML para automatizar la selección de características, algoritmos y optimización de hiperparámetros.
- **Desarrollo colaborativo**: Adoptar sistemas como Azure DevOps para gestionar el código y pipelines colaborativamente entre equipos de datos.

3. Acceso y reentrenamiento de modelos

El aumento en la cantidad de usuarios y datos requerirá:

- **Escalado de APIs:** Usar Azure Kubernetes Service (AKS) o Azure App Service con balanceo de carga para manejar un alto volumen de solicitudes.
- **Reentrenamiento automatizado**: Diseñar triggers basados en la llegada de nuevos datos o en la degradación del desempeño del modelo, como ejemplo:
    - Monitorear métricas como AUC-ROC o precisión en producción.
    - Usar pipelines de CI/CD para desplegar modelos actualizados sin interrupciones.

4. Pipeline de DevOps

El pipeline de DevOps debe ser más robusto y capaz de manejar múltiples modelos y actualizaciones frecuentes:

- **Automatización avanzada:** Usar herramientas como GitHub Actions o Azure Pipelines para integrar pruebas unitarias, pruebas de integración y despliegues automatizados.
- **Monitoreo en tiempo real:** Implementar Azure Monitor y Log Analytics para rastrear métricas clave como uso de CPU, latencia de API y fallos en tiempo real.
- **Seguridad:** Incorporar autenticación y autorización avanzada (por ejemplo, Azure Active Directory) para controlar el acceso a APIs y recursos.

Nombre: Ángel Fernández Galán
Dni: 08896350L

En resumen, con una fuerza laboral de 1000 personas, el pipeline de DevOps, la estrategia de análisis, el desarrollo de modelos de ML y el acceso a los modelos deben diseñarse para manejar la mayor escala, el volumen de datos y los requisitos computacionales. Escalar la infraestructura, implementar marcos de cómputo distribuido y optimizar los pipelines para la automatización y eficiencia serían cruciales para satisfacer eficazmente las necesidades de la plantilla más numerosa. De manera más importante, quizás, una plantilla mucho más numerosa conllevaría la posibilidad de desarrollar modelos más precisos.

4 - *Productivizar el modelo de bank marketing. Hay dos opciones para elegir: mediante el diseñador de Azure o bien de forma manual en su propia máquina (serializando los modelos/transformaciones con Pickle).*

1. *Si el objetivo del modelo es lograr predecir previo a que se realice la llamada qué personas tienen una alta probabilidad de contratar, ¿se podría decir si hay alguna variable que no sea posible incluir en el modelo en producción?, ¿Cuáles son?*

La variable 'duration' está altamente relacionada con el target output (e.g. si duration=0, entonces y=no). La duración no se conoce antes de la llamada, y después se sabe la duración exacta. Como indican los autores del dataset, esta variable solo debería ser usada en benchmarking. Se elimina también la variable 'pdays', puesto que contiene taje muy elevado de una sola categoría ('-1') y su poder decisorio es muy bajo, según el PCA realizado.

2. Probar varios modelos e ingeniería de características y dar con el modelo que mejor métrica presente.

Se adjunta un documento (ModeloML.pdf) detallando la elaboración del modelo y la ingeniería de características, así como el despliegue del endpoint.