/bluetab
an IBM Company

Aumentando las capacidades de uso de la Inteligencia Artificial en la organización

01

Introducción

En todas las industrias, hemos visto como DevOps y DataOps han sido adoptadas ampliamente como metodologías para mejorar la calidad y reducir time-to-market de las iniciativas relativas a la ingeniería de software y a la ingeniería de datos respectivamente. Sin embargo, el debate sobre MLOps suele centrarse exclusivamente en las herramientas, obviando que un aspecto crítico del éxito de la inversión en Machine Learning (ML) es permitir que las personas puedan lograr sus objetivos, lo que implica diseñar la estructura adecuada para su organización.

Además, hay que tener en cuenta que los aspectos únicos del machine learning, y cómo los principios generales de desarrollo de software no siempre pueden ser aplicables a estos proyectos.

¿Por qué Bluetab piensa que esto es importante?

02

Diseño de la Organización

La estructura organizativa influye en cómo de bien se alinean las personas para lograr los objetivos de negocio. Según la madurez de su organización, los perfiles de ML pueden integrarse en equipos comerciales en lugar de centralizarse en equipos aislados. Definir roles claros es fundamental para que las personas se concentren en áreas de interés sin estar demasiado dispersos; algunas empresas también pueden beneficiarse de un equipo dedicado, centrado e en administrar la plataforma de ML

A continuación se definen una serie de competencias, con los títulos de trabajo de referencia:

- **Data Scientist**: Explora y encuentra ideas que influyan en las decisiones de negocio.
- **Machine Learning Engineer**: Desarrollar y evaluar modelos de ML.
- **Machine Learning Researcher**: Experimentar con nuevas arquitecturas de ML..
- **Data Platform Engineer**: Gestiona la infraestructura de datos.
- **MLOps Engineer**: Gestiona las plataformas de ML y las operaciones.
- **Machine Learning Auditor**: Controla los riesgos asociados a las soluciones de ML.
- **Product Manager**: Define qué soluciones de ML se deben construir.
- **Software Engineer**: Integra las capacidades de ML en las aplicaciones existentes.

Un componente clave en la eficiencia de la organización es formalizar cómo se intercambia la información con el fin de impulsar la alineación con los objetivos y minimizar la duplicación de esfuerzos no solo influye las tecnologías o plataformas, sino también se necesita definir aquellos procesos que son fundamentales para el éxito de los proyectos de ML.

Procesos

Evaluar qué proyectos de ML maximizan el impacto en el negocio no suele ser una tarea principal que forme parte de los proyectos de ML. El impacto empresarial puede adoptar muchas formas: reducir riesgos, aumentar el impacto en ESG o, lo que es más obvio, mejorar los ingresos o reducir los costes operativos.

Elegir los proyectos adecuados y evaluar el valor empresarial por adelantado es fundamental para el éxito a largo plazo. Como referencia, tenemos la checklist[1] publicada por fast.ai donde encontraremos los aspectos más importante a la hora de decidir qué proyectos de ML priorizar. Sin embargo, la mayoría de los proyectos de ML fracasan por culpa de estructuras y procesos organizativos deficientes. A veces, el valor de un proyecto no justifica su coste, puede que los datos no estén disponibles o, puede que las malas prácticas de desarrollo de software ralentice la experimentación y dificulten la colaboración. Lo que equivale a decir que hay muchas razones por las que los proyectos deben implementar una visión MLOps para minimizar los riesgos operativos.

El gobierno de MLOps propone la definición de 7 procesos con diferentes grado de madurez para lograr tener una plataforma integra de Machine Learning:

- **Development**: este proceso trata de identificar si existen una cantidad de datos suficientes, así como definir las tareas de preparación y validación para garantizar la calidad de estos datos, antes de la experimentación.
- **Operativización**: automatizar el proceso de training y testing en base a pipelines repetibles y fiables.

[1] https://www.fast.ai/posts/2020-01-07-data-questionnaire.html

- **Continuous training**: ejecutar el proceso de training en respuesta a nuevos datos o cambios del código, o en base a un horario, potencialmente con nuevas configuraciones de entrenamiento.
- **Model deployment**: define un proceso de despliegue hacia la infraestructura de serving, ya sea en batch y/o online.
- **Prediction serving**: se trata de establecer la infraestructura que implementa la producción para la inferencia.
- **Continuous monitoring**: consiste en un proceso para monitorear la eficacia y la eficiencia de un modelo, así como la calidad de los datos, en el tiempo.
- **Management**: es el proceso centralizador del gobierno que permite respaldar la auditoría, trazabilidad y cumplimiento, y promover la colaboración.

En la siguiente sección compartiremos las claves para definir cada uno de los procesos..

04 MLOps: The Hard Way

Antes de embarcarse en un proyecto de ML, con frecuencia los datos no existirán en una tabla SQL fácil de recuperar, o es posible que se desconozca su estado, al igual que el progreso del proyectos tener avances inesperados. Esta incertidumbre no solo afecta sobre cuánto tiempo pueden requerir las tareas específicas, sino también sobre qué tareas realizar. Lo importante es tener claro cómo estimar el impacto comercial o los plazos de por adelantado, atendiendo a la necesidades que implican un cambio organizacional.

Proceso de desarrollo en ML

En una fase temprana del proceso, el objetivo del data-science es probar varias ideas rápidamente mediante exploración de datos, además de gestionar las expectativas al aplicar algoritmos basados en inteligencia artificial. También es importante que la fase de experimentación sólo debe empezar cuando el caso de uso de ML está bien definido, lo que significa que está bien documentado y se han respondido las siguientes cuestiones:

¿Cuál es la tarea final?
¿Cuál es la métrica de evaluación?
¿Qué datos son los relevantes?
¿Cómo se puede medir el impacto comercial?
¿Qué necesidades de consumo tiene?

Los data-scientist deben usar en la fase de experimentación para obtener un prototipo de modelo suficientemente efectivo para el caso de uso que tenemos definido. Como hemos introducido, los data-scientist tienden a implementar un sistema end-to-end, pero la

plataforma de ML necesita administrar la ejecución de los componentes que proponen los equipos de producto . Estos componentes son artefactos software que se organizan en una pipeline que simulará el trabajo de programación de un data-scientist. Adicionalmente, mediante la fase de experimentación se definen las siguientes tareas:

(1) Descubrimiento de datos disponibles, analizando su naturaleza y seleccionado aquellas variables que identifican las expectativas.
(2) Preparación de datos realizando las modificaciones que se estimen oportunas para cada una de las variables seleccionadas mediante herramienta de prototipo rápido o código.
(3) Selección del modelo que encaje con las propiedades de nuestras variables.

Finalmente, los data-scientist empezarán a refinar la solución al caso de uso planteado mediante un proceso iterativo hasta encontrar el impacto comercial.

Proceso para industrializar el entrenamiento

Antes de poner en funcionamiento cualquier modelo de ML, debemos realizar experimentos para validar su viabilidad y estimar su impacto en el negocio. La operativización está relacionada con preparar los activos que definen los workflow compartidos en la organización que permiten a los data-scientist iterar en sus ideas de forma eficaz y sencilla, siendo esta tarea una de las principales de los ML engineers que dan soporte a la plataforma de ML.

Un proceso sólido de desarrollo y evaluación de modelos permitirá a los data-scientist iterar rápidamente, comunicarse de manera efectiva con las partes interesadas y evaluar adecuadamente los modelos. Los ML engineers deben ser capaces de integrar un sistema que facilite estas tareas, lo que significa que al menos se definen los siguientes aspectos:

(1) Definir cuál va ser el entorno objetivo.
(2) Accesos a los orígenes de datos en tiempo de ejecución.
(3) Permisos adecuados para ejecutar el código.
(4) Evaluar las métricas de calidad de los modelos.

Con frecuencia, es necesario mirar las métricas no solo en conjunto, sino también en segmentos específicos de datos, lo cúal requiere de un paso previo de configuración un

entorno de desarrollo, que implica configurar una infraestructura remota con gran capacidad de cómputo en lugar de propios equipos personales. Jupyter Notebooks es fundamental para crear rápidamente prototipos y experimentar con ideas, por lo cual cualquier plataforma de ML debe estar adaptada para su uso..

Proceso de Continuos Training

A diferencia de los dos primeros procesos, donde se describen tareas a realizar por personas, el proceso de continuos training consiste en orquestación y automatización de las pipelines de entretenimiento a nivel de infraestructura, definiendo cual es la frecuencia con la que vuelve a realizar el entrenamiento de un modelo, dependiendo de las reglas y el impacto en costes del caso de uso. Aunque este es el final del proceso de desarrollo, también es el comienzo del ciclo de mantenimiento del modelo.

Los modelos deben ser supervisados y periódicamente re-entrenados para corregir problemas de eficiencia, aprovechar nuevas variables de entrada o simplemente a cambios en el código. Este proceso realmente tiene que ver con una iteración sistemática para refinar y asegurar continuamente la precisión del modelo. Sin embargo, identificar el momento adecuado para volver a ejecutar un entrenamiento para un modelo no es una tarea trivial: volver a entrenar con demasiada frecuencia puede significar interrupciones en el servicio sin significativas nuevas mejoras, mientras que no volver a entrenar con suficiente frecuencia puede llevar a la degradación del rendimiento del modelo.

Cada ejecución del pipeline de continuos training puede ser desencadenado de varias formas, incluidas las más simples como las siguientes:

> A. Trabajos programados en función de la configuración que se desee (por ejemplo, tiempo)
> B. Ejecuciones basadas en eventos, como cuando nuevos datos están disponibles por encima de un cierto umbral que modificara la forma de la distribución de los datos.
> C. Cuando se detecta un deterioro sustancial en el proceso de monitorización se pueden realizar invocaciones manuales gestionadas por el equipo de la plataforma de ML.

Las típicas tareas de este tipo de procesos incluyen los siguientes workflows:

1. **Data Ingestión.** Los datos de entrenamiento se extraen del conjunto de datos de origen y del repositorio de características utilizando los criterios de extracción definidos por los data-scientist (aka queries) y el periodo de la actualización más reciente.
2. **Data Validation**. Los datos de entrenamiento que han sido extraídos se validan para asegurarse de que el modelo no esté entrenado usando datos sesgados o incompletos.
3. **Data Transformation:** Los datos se dividen, normalmente en divisiones de entrenamiento, evaluación y validación, siendo entonces transformados para obtener las características que se diseñaron según lo esperado por el modelo.
4. **Model Training.** El algoritmo se entrena y los hiper-parámetros se ajustan durante cada una de las iteraciones del entrenamiento para producir el mejor modelo posible.
5. **Model Evaluation**. El modelo es evaluado contra los datos de test para obtener el rendimiento teórico, empleando diferentes métricas y utilizando diferentes estrategias de particionado de datos.
6. **Model Validation**. Los resultados de las evaluaciones previas del modelo se refutan para asegurarse que el modelo cumple con los criterios de negocio.
7. **Model Registry**. El modelo validado se almacena en el registro de modelos con los metadatos necesarios para hacerlo funcionar.

Por último, como veremos en el proceso de continuos monitoring, promover un modelo a producción requiere seguir un flujo de trabajo definido, y además, necesitamos poder registrar el linaje del modelo para poder asociar datos a métricas que van apareciendo como veremos más adelante.

Proceso de Despliegue de Modelos

Después de entrenar, validar y añadir al model registry un modelo candidato, por fin estamos listos para su despliegue final. Durante el proceso de despliegue tiene lugar el empaquetado del software definido, testeo de su correcto funcionamiento y despliegue en un entorno con el objetivo de dar servicio.

La parte de despliegue continuo, es similar a la entrega progresiva que se puede encontrar en la metodología DevOps. Este tipo de despliegues se realizan mediante la ejecución de una estrategia tipo canary o blue-green. que se centran en la eficiencia del propio proceso de

serving, evitando cualquier error que provoque una pérdida de servicio. La obtención de predicaciones online es un hito particularmente importante en el contexto de ML. Decidir si un nuevo modelo candidato debe reemplazar al modelo de producción es más complejo en comparación a una tarea clásica de la ingeniería de software.

En enfoque de la entrega progresiva, el nuevo candidato no sustituye inmediatamente a la versión anterior, sino después de cierto tiempo en que ambos modelos conviven en paralelo en producción. Un subconjunto de los consumidores es redirigido al nuevo candidato, incrementando el tráfico en varias etapas hasta que el resultado final decide que el modelo es liberado completamente y sustituye al anterior. Para esta tarea, las pruebas A/B son útiles y se pueden utilizar para cuantificar el impacto del nuevo modelo en los objetivos del caso de uso y las aplicaciones que los consumen.

Proceso de Serving

El proceso de serving de las inferencias empieza justo después de que el modelo haya sido desplegado en el entorno de producción. El modelo comienza a aceptar peticiones de información sobre los datos y genera las respuestas adecuadas.

El sistema de serving se puede ofrecer con las siguientes formas de consumo:

- **Inferencia online en tiempo real para las tareas con una alta frecuencia utilizando endpoints.**
- **Streaming en aquellos casos que lleguen los eventos a sistemas de colas o buses.**
- **Batch por lotes offline, generalmente integrado en procesos masivos de ETLs**
- **Inferencia integrada como parte de sistemas IoT o dispositivos edge.**

Todos los registros producidos en las inferencias de las predicciones y otras métricas de consumo deben ser almacenadas para su posterior análisis y seguimiento, como veremos en el siguiente proceso de monitorización continua.

Proceso de Continuos Monitoring

La monitorización continua es vital en cualquier plataforma, y la supervisión de los modelos es área crucial de la metodología MLOps para el control de la eficiencia y la degradación en producción. Este proceso verifica de forma regular y proactiva el

rendimiento del modelo para evitar cualquier incidencia en el rendimiento debido a que a medida que las peticiones tienen nuevos datos que varían su distribución con el tiempo, haciendo que sus propiedades comienzan a desviarse de las referencias que se usaron para entrenar y evaluar el modelo.

Este proceso conduce a un modelo más efectivo en el tiempo al evitar la degradación, además, las modificaciones en los sistemas complementarios que realizan la transformación u obtención de la información de la peticiones puedan producir cambios en las variables que en consecuencia también pueden producir malas predicciones a los sistema de inferencia, por tanto la monitorización se utiliza para hallar en los registro cualquier indicativo que puede identificar anomalías, sesgos o valores atípicos. Un proceso tienen que constar al menos con los siguientes pasos para que sea efectivo:

1. Una muestra de las peticiones y respuestas debe ser extraída del servicio de almacenamiento central de logs.
2. El sistema periódicamente carga las inferencias más recientes para calcular las estadísticas de las predicciones que el modelo está produciendo.
3. Los datos generados son comparados con la referencia para evidenciar los posibles sesgos, además de comparar las estadísticas calculadas con las extraídas de los datos de evaluación.
4. Si existen predicciones supervisadas con intervención manual para los datos publicados, el sistema puede evaluar la efectividad de las predicciones.
5. Si finalmente se identifica algún tipo de degradación en la eficiencia, se generan las alertas que puede enviar por diferentes medios de notificación para desencadenar un nuevo ciclo de re-entrenamiento.

En particular, cuando hablamos de que el modelo está sufriendo de una degradación de la eficacia esperada, en realidad se tiene que definir en términos de data y concept drift. El primero describe el distanciamiento creciente en el conjunto de datos que se utilizó para entrenar, evaluar y validar el modelo y las variables reales que está usando el modelo para

producir las precisiones, mientras que el segundo radica el cambio en las relaciones entre los datos de entrada y salida en el caso de uso.

Cuando hablamos de data drift, estamos involucrando a dos aspectos clave

> Modificaciones en el esquema que ocurren cuando los datos de entrenamiento y de serving no comparten la misma estructura.
> Las distribuciones de las variables que forman las características para el entrenamiento es significativamente diferente a la distribución que obtiene la producción de predicciones.

Adicionalmente a las técnicas anteriores, otras técnicas pueden detectar también desviaciones en los datos que incluyen la detección de valores atípicos en los datos, cambios en la función de las variables, o connotaciones éticas o de género en los modelos productivos. Igualmente, en algunos escenarios es viable añadir información que es capturada desde el back-office de la organización, utilizando el repositorio de característica para almacenarla y que sea empleada en el futuro durante un nueva ejecución del proceso de continuos training.

Además de medir la efectividad, el propio servicio de infraestructura tiene que registrar la actividad del consumo de recursos mediante las siguientes métricas:

> Uso de los recursos de cómputo, incluyendo CPUs, GPUs, y memoria.
> La latencia de generación de las predicciones para indicar la salud del servicio.
> Throughput que es una métrica general a cualquier despliegue.
> Tasas de error producidos en el serving.

Cuantificar estas métricas no solo es útil para mantener y mejorar el rendimiento global del sistema, sino también en la gestión de costes.

05

La plataforma

A menudo es difícil desacoplar los procesos de la tecnología. En un mundo ideal, empezaremos por el proceso y elegimos la tecnología más adecuada para implementarlo, pero en la realidad, la tecnología influye en nuestros procesos y los limita. Merece la pena construir una plataforma de ML suficientemente flexible para dar soporte a workflows, que a su vez se integren con varias soluciones tecnológicas diferentes.

Desde el punto de vista tecnológico, tenemos 3 indicadores que facilitan tomar las decisiones más relevantes al construir su plataforma de ML.

1) **Frameworks**

   Cada vez son más las herramientas low-code o no-code ganan más terreno en la plataformas de ML, pero los profesionales mantienen su preferencia por el código (PyTorch, Tensorflow, Keras, Scikit-learn, etc.).

2) **Ops tools**

   Realizar cada tarea de forma manual no es eficiente, se tiende a adoptar herramientas de desarrollo que faciliten las mejores prácticas y obtengan los mejores resultados. Herramientas como Kedro, MLflow o Weights&Biases ayudan al seguimiento de los modelos, registro y etiquetado, versionado, supervisión, y la orquestación final.

3) **Infraestructura**

   Con el fin de ejecutar los workflows, se necesita de una gran cantidad de recursos de procesamiento y almacenamiento, que tiene una demanda elástica en el tiempo, luego se suelen obtenerse a través de la computación en nube (AWS, GCP, Azure), aunque algunas organizaciones siguen optando por gestionarlas por su cuenta.

Es importante tener un equilibrio entre las tareas de los diferentes roles de la organización, cuando más se preocupen los data-scientist por un proceso en particular, más libertad querrán. Por norma general, crear una plataforma de ML que gobierne esta flexibilidad es una buena idea, proporcionar soluciones de alto nivel que les permiten reducir el tiempo dedicado a configurar los recursos de infraestructura les limitará la libertad, pero focalizará en los tareas concretas donde más valor pueden aportar. Por último, es importante conocer a los desarrolladores, ofrecerles un entorno familiarizado con APIs, Python o SQL, y que se apoyen en entornos de desarrollo donde se sientan cómodos como Jupyter Notebooks.

06

Nuestro punto de vista

AI está experimentando una ola de innovación sin precedentes

Estamos viviendo una oleada de innovación sin precedentes en el mundo de AI. En una encuesta realizada por McKinsey el pasado mes de agosto titulada "The state of AI in 2023: Generative AI's breakout year"[2] 1/3 de los encuestados indican que en sus organizaciones se está utilizando de forma regular Generative AI para alguna tarea. ¡Generative AI ha eclosionado hace apenas 12 meses!

¿Cómo es eso posible?

AI es una tecnología especial. Una forma de visualizar AI es a través de una gran caja de herramientas (los modelos). Estas herramientas son de diferentes tipos como aprendizaje supervisado, Generative AI, o aprendizaje reforzado y valen para realizar una gran variedad de tareas como convertir una imagen en texto, identificar spam en el email o traducir. Esta variedad de tareas hace que el potencial de transformación de AI sea muy alto.

Una de las características de esta ola de innovación es la aparición de nuevos tipos de herramientas muy potentes. Generative AI ha añadido nuevas capacidades a la caja de herramientas como resumir conversaciones, escribir emails o gestionar el conocimiento, y tiene unos ciclos de desarrollo mucho más rápidos. ¡De meses a semanas!.

---

2 https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2023-generative-ais-breakout-year

Las herramientas más habituales de AI son las de aprendizaje supervisado. Estas herramientas son muy buenas para reconocer patrones, pero crear una herramienta de este tipo requiere mucho tiempo. Hay que preparar un conjunto muy grande de datos de entrenamiento y realizar el entrenamiento. Este proceso suele ser de más de 6 meses. Generative AI permite crear herramientas de una forma mucho más rápida. Partimos de una versión de la herramienta pre-entrenada, y sólo tenemos que adaptarla utilizando un conjunto pequeño de datos. El proceso es de algunas semanas.

En este escenario de tener una caja de herramientas cada vez con muchas más capacidades y con ciclos de desarrollo más rápidos, es normal que los CEOs y los CIOs de las empresas tengan AI en sus agendas estratégicas.

Instalar elementos y prácticas de gobierno de la AI es uno de los elementos claves para tener éxito en esta aventura

Adoptar AI en una empresa, hacer crecer su caja de herramientas, se ha acelerado mucho. AI final lo que se busca es transformar la empresa haciendo que sus actividades sean más eficientes y productivas, o creando actividades nuevas.

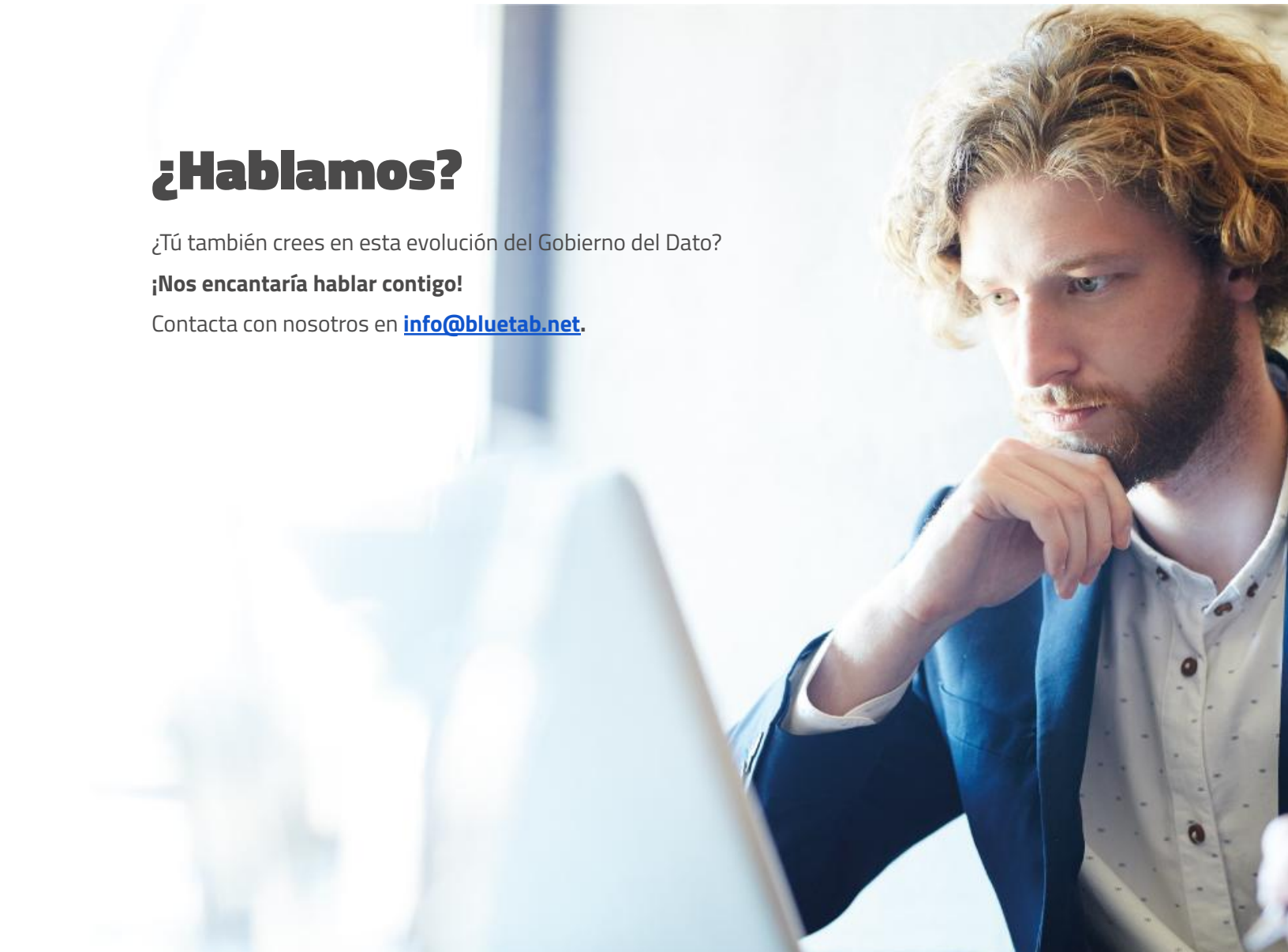
Usar AI incorpora riesgos como sesgos, falta de precisión o falta de equidad. AI está mejorando cada vez más y estos riesgos se están reduciendo, pero siempre hay que considerarlos. De igual forma que hay que considerar siempre el uso ético de esta tecnología.

Una buena idea para tener éxito en la adopción de la AI es instalar desde el principio elementos y prácticas de gobierno de la AI, y mejorar esos elementos y prácticas de gobierno de la AI al mismo tiempo que aprendemos a usar la AI.

El gobierno de la AI abarca desde la gestión de la caja de herramientas, el entrenamiento, el análisis de los riesgos, la medición de los impactos, las formas de trabajo en la plataforma de datos y AI, la monitorización hasta las decisiones sobre priorizar casos de uso, invertir en capacidades o establecer los principios éticos sobre el uso de AI en la empresa.

Nosotros entendemos el gobierno de la AI como un camino con una curva de madurez. Por eso vemos clave instalar sus elementos y prácticas desde el principio.

Otra pregunta interesante de la encuesta de McKinsey "The state of AI in 2023: Generative AI's breakout year" es sobre cuáles son los elementos que están suponiendo un mayor desafío a la hora de conseguir los beneficios de AI. Los encuestados calificados como "high performers" señalan elementos relacionados con su madurez en el uso y gobierno de la AI como monitorización del rendimiento de las herramientas y re-entrenamientos, mientras que el resto de encuestados señalan elementos más fundacionales como la estrategia y el talento.



¿Hablamos?

¿Tú también crees en esta evolución del Gobierno del Dato?

¡Nos encantaría hablar contigo!

Contacta con nosotros en **info@bluetab.net**.


/bluetab
an IBM Company