# **Primera clase de optimización**

Analítica prescriptiva

Es la tercera etapa de madurez de la analítica y quizá la más abandonada. Siempre nos enfocamos en los datos, pero no siempre nos preguntamos para qué los tenemos. Esa es la pregunta que busca responder este tipo de analítica. Está orientada hacia la toma de decisiones no triviales. ¿Qué es lo que me interesa?

La analítica de negocios es el proceso científico de analizar y transformar datos con el fin de hacer decisiones óptimas. Esto último la diferencia. No nos vamos a centrar tanto en los datos sino en la forma en la que los analizamos. El para qué.

El proceso de toma de decisiones con BA sigue:

1. Tomar visiones del sistema y extraer datos (con este insumo no contaba la IO tradicional). Hoy en día, nos damos cuenta que nuestras decisiones no habían sido tan racionales como pensábamos. Los datos nos han permitido esto y muchas veces nos comunican la irracionalidad de los patrones analizados.
2. Analizar los datos.
3. Pronosticar
4. No lo copié
5. No lo copié

Los métodos analíticos se categorizan:

**Analítica descriptiva** 🡪 **Analítica predictiva** (regresión, series de tiempo, data-mining, simulación)🡪 **Analítica prescriptiva** (simulación, optimización, análisis de decisión).

Argumento principal de la optimización 🡪 El pasado se repetirá en el futuro, desde una perspectiva estadística. Por tal motivo, sólo son útiles para nosotros en el corto plazo. La planificación por escenarios es más adecuada para el mediano-largo plazo. ¿Será posible hablar de mega tendencias? No es válido predecir tanto tiempo, por ejemplo, 15 años.

Se estima que sólo el 3% de las empresas del mundo utiliza herramientas de analítica prescripta para tomar sus decisiones.

R y Python no son programas adecuados para OPTIMIZAR.

Las revistas Analytics e Interfaces son las más apropiadas para el trabajo del curso de optimización.

IMPORTANTE 🡪 En optimización, debo usar esquemas diferentes, dependiendo de si estoy respondiendo una pregunta discreta o continua.

Coeficientes tecnológicos 🡪 lo que una máquina consume en términos de recursos.

Programación lineal, no lineal y entera

En PL, siempre la solución óptima está en un vértice. En PNL, no necesariamente es así.

En PL, los resultados que obtengo pueden ser enteros, pero no necesariamente es así. Esta restricción de enteros si es de obligatorio cumplimiento en la programación entera PE.

En la programación lineal, existen dos algoritmos básicos: el método simplex y el algoritmo de punto interior, que fue desarrollado por IBM y que les permitió ganar mucho dinero porque se mantuvo oculto mucho tiempo. Aun hoy, muchos programas no revelan sus algoritmos para solucionar problemas de optimización.

Usar PL siempre es más fácil. En la diapositivas están expuestos algunos métodos para linealizar algunas funciones objetivo que pueden aproximarse a una recta.

En general, los algoritmos de PNL, se confunden fácilmente buscando las soluciones óptimas. Muchas veces encuentran como solución óptima global, un óptimo local, pues se engañan ante puntos cuya derivada es cero.

Función objetivo mini-max(minimizar un máximo) 🡪 Propio de teoría de juegos. No la entienden los algoritmos de optimización lineales.

En general, los software de optimización utilizan diferentes algoritmos analíticos, lo más comunes son:

**PL 🡪 Simplex**

**PNL 🡪 relejación Lagrangeana**

PEM 🡪 ramificación y acotamiento. Encontrar soluciones a problemas de este tipo es muy complejo. Hoy en día, los algoritmos genéticos han facilitado estos procedimientos.

# **¡Los problemas de programación entera son muy difíciles de resolver!**

En este contexto, analíticos quiere decir que se basan en formulaciones estrictamente matemáticas.

La nueva tendencia en optimización, nos habla de técnicas metaheurísticas, que se basan en inteligencia artificial y no tienen una comprobación matemática. Adicionalmente, cuando se tienen problemas muy complejos, puede optarse por la descomposición de los mismos 🡪 Por ejemplo, en una compañía, puedo asignar diferentes funciones objetivo a los departamentos, para conseguir un objetivo global (problema máster). Esto no lo hacen los computadores, sino que debe planificarse estratégicamente primero.

En general, los problemas de PL encuentran regiones factibles CONVEXAS. Esto quiere decir, que cuando encontramos un punto óptimo, los valores anteriores y posteriores son inferiores, y están en la misma región factible. Por lo tanto, no tengo que seguir buscando intensamente otra solución mejor. A esto, se le suma la ventaja de que vamos a encontrar la solución óptima en un vértice. Por lo tanto, siempre tratamos de linealizar todos nuestros problemas.

A pesar de que la PL no soluciona todos los problemas de gran complejidad, si ha sido muy útil por más de 60 años. Se dice incluso que salvó la II Guerra Mundial, cuando se reunió a los grandes matemáticos de la época. De allí, salió el nombre de la simulación de Montecarlo y el método simplex 🡪 ¿Qué pasaría si los misiles caen en los puntos x para el enemigo? ¿Cómo maximizar el daño?

Análisis de sensibilidad y teoría dual

# **Precios sombra de los recursos**

Hemos aprendido a través de los modelos de optimización, que la vida es determinística, pero en realidad no es así. Siempre que modelamos, debemos tener en cuenta que los pronósticos pueden ser muy exitosos o fallar por completo. Por lo tanto, analizar la sensibilidad de mi modelo y el comportamiento del sistema que estoy representando, siempre debería ser fundamental, para que lo que planificamos, no termine siendo un desastre y se mantenga la optimalidad.

Sin embargo, no podría estar preguntándome: ¿Qué pasaría si…? Y luego optimizo cada uno de esos escenarios. No es eficiente.

Para lidiar con esta dificultad, el método simplex inventó la siguiente metodología:

* En lugar de cambiar los valores de las restricciones o la función objetivo, determino lo que se llama los “precios sombra”, que me indican el cambio en mi función objetivo, cuando se genera un aumento en una unidad de recursos disponible.
* Reformulo el problema con mis nuevos valores (valores originales + precio sombra) y optimizo.

Lo anterior responde a las preguntas: ¿Estoy siendo eficiente con mis recursos? (para el caso de minimizar costos) y ¿Cuánto debo cobrar?. Por ejemplo, para el caso de los vestidos (demanda), el precio sombra me indica cuál es el precio justo que debe pagarse por un vestido.

Del lado de los recursos 🡪 Precio sombra me indica cuánto puedo ganar de más

Del lado de la demanda 🡪 Cuánto debo cobrar por mi producto o servicio.

# **Costos reducidos/marginales de los coeficientes de la función objetivo**

Es otra forma de realizar análisis de sensibilidad. Tiene que ver con los costos cuando trato de maximizar utilidad y con las utilidades cuando quiero minimizar costos.

Que me dice un costo marginal 🡪 si una variable de decisión es cero, cuánto debo incrementar mis coeficientes para que justifique la ganancia. Es válido cuando el problema o la situación exigen que ese recurso que salió cero desde la perspectiva matemática, tiene que utilizarse, sea porque ya se contrató o porque no es válido renunciar a él.

# **Estrategia del problema maestro y el subproblema**

Para que la conversación entre los óptimos locales y el global no redunde en una referencia circular (la solución de uno, es el insumo del otro), el subproblema no va a pasar una solución al problema maestro, sino una recomendación, en términos de los PRECIOS SOMBRA, para que él decida que escoger en un intervalo de soluciones más flexible.

Por ejemplo, en un problema de planificación multiperíodo, el problema maestro termina solucionándose por convergencia, es decir, cuando todas las soluciones locales empiezan a “opinar” lo mismo y ya no es necesario hacer cambios. Normalmente, alcanzar este estado de convergencia requiere de 20 a 30 corridas del modelo.

¿Cuántas variables de decisión tener? Si son continuas, puedo tener hasta +70.000 sin problema. Pero, si intervengo variables enteras o binarias, 700 de ellas ya pueden representar un gran reto para un optimizador.

# **Programación binaria**

Me pregunta por cuáles proyectos emprender.

# **Programación mixta**

Combina programación binaria y entera (porque también me pregunta cuántos).

# **Programación entera mixta**

Formulación problemas de los hubs de aerolíneas

SET 🡪 S(i): las ciudades que están a menos de 1000 millas de la ciudad i

Xi = 1 🡪 Si se instala un hub en la ciudad i

Xi = 0 🡪 En caso contrario

F.O 🡪 Min

Restricciones

S(Atlanta): X1 + X3 + X5 + X7 + X8 + X9 >= 1 🡪 Todas estas ciudades me atienden a Atlanta

S(Boston): X2 + X8 + X9 >= 1

Y así sucesivamente para todas las ciudades en su orden.

De forma genérica sería:

**IMPORTANTE: debemos crear un alias de i, que será “j” y será el mismo subconjunto de i. Esto para que el software no se confunda.**

Finalmente,

Bin(Xi) 🡪 para indicar que es binaria

Próxima clase veremos cómo introducir esto en software.

TAREA: resolver problema de ubicación de CAI)

Xij = 1 🡪 Si la estación de policía i, atiende a la zona j.

Xij <= Yi 🡪 Restricción fundamental (Sólo puedo atender una zona j, si instalo la i)

Yi 🡪 personas atendidas o zonas cubiertas