



**Diagnóstico e implementación de mejoras en un sistema optimizador de margen bruto
para empresa productora de alimentos.**

Santiago Ruiz González

Informe de práctica como requisito para optar al título de:
Ingeniero Electrónico.

Asesor Interno
David Stephen Fernández Mc Cann
Profesor Facultad de Ingeniería
Universidad de Antioquia

Asesor externo
Natalia Romero Ríos
Analytics and Modeling Associate Manager
Accenture Colombia

Universidad de Antioquia
Facultad de Ingeniería

Departamento de Ingeniería Electrónica y Telecomunicaciones
Medellín, Colombia
2022

Cita	(Ruiz González, 2022)
Referencia	Ruiz González, S., (2022). <i>Diagnóstico e implementación de mejoras en un sistema optimizador de margen bruto para empresa productora de alimentos</i> , Semestre de industria, Universidad de Antioquia, Medellín.

Estilo APA 7 (2020)



Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/director: Jesús Francisco Vargas Bonilla.

Jefe departamento: Augusto Enrique Salazar.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Tabla de contenido

Tabla de contenido.....	3
1. Resumen	4
2. Introducción.....	4
3. Objetivos.....	5
3.1. Objetivo General	5
3.2. Objetivos Específicos.....	6
4. Marco Teórico	6
4.1 Método Delphi.....	6
4.2. Análisis de datos.....	7
4.3. Regresión <i>ElasticNet</i>	7
4.4. Regresión Lineal Múltiple.....	8
4.5. Framework de desarrollo de aplicaciones web <i>Dash</i>	8
4.6. Servicios de Amazon AWS	9
5. Metodología.....	9
5.1. Diagnóstico del sistema de optimización de margen bruto mediante el método Delphi.	9
5.1.1 Aplicación del método Delphi	10
5.1.2 Elementos del sistema a mejorar	10
5.2. Ajuste de algoritmos usados y modelos predictivos.....	11
5.2.1 Actualización de los modelos para 2 de los SKU's incluidos	11
5.2.2 Modificación de los algoritmos y mejora en la eficiencia del código.....	11
5.3. Implementación de una nueva versión del dashboard usando el framework Dash.....	11
5.3.1 Modificación de la cantidad de pestañas del dashboard.....	12
5.3.2 Modificación de sección de optimización y simulación	13
5.3.3 Creación de ventana de autenticación.....	16
5.3.4 Modificación del tutorial del dashboard	18
5.3.5 Despliegue en el servicio AWS Elastic Beanstalk.....	19
6. Resultados y análisis	19
6.1 Análisis de las métricas de desempeño de los modelos	19
6.2 Análisis del tiempo de respuesta en los elementos del dashboard	20
7. Conclusiones	21
8. Referencias bibliográficas.	21

1. Resumen

Actualmente los datos se han convertido en una herramienta fundamental para lograr que las empresas puedan acceder a información que aporte valor y les facilite tanto la toma de decisiones como la generación de estrategias de negocio. Hay compañías como Accenture que brindan capacidades de analítica avanzada a sus clientes para que generen valor a partir de sus datos y logren anticiparse a eventos futuros tomando las mejores decisiones. Los clientes de esta compañía son de diversos sectores del mercado, entre ellos se encontraba una empresa productora de alimentos, que buscaba obtener el valor adecuado de precio y descuentos para sus productos, optimizando sus ganancias.

De allí surgió el caso de uso asociado a la generación de un sistema de optimización de margen bruto, con el que se buscaba obtener la mejor combinación entre precio por unidad de los productos y descuentos, el cual se implementó de manera parcial, por lo que era necesario realizar un diagnóstico para identificar las oportunidades de mejora de este sistema. Este diagnóstico se llevó a cabo de la mano del equipo de analítica avanzada y el cliente haciendo uso del método Delphi, y una vez identificadas estas falencias se implementaron para tener un sistema más eficiente que garantizara la coherencia de los resultados. Estas mejoras realizadas permitieron cuantificar la reducción del tiempo de cargue de la información, observando una reducción significativa y asimismo se logró que los resultados presentados de los modelos fueran coherentes y confiables.

2. Introducción

El análisis de datos es una ciencia de gran interés y utilidad en la actualidad gracias a sus importantes aplicaciones en diversas industrias y áreas del conocimiento [1]. Esta ciencia se encarga de examinar un conjunto de datos con el objetivo de obtener información adicional y extraer conclusiones que puedan ser de utilidad a la hora de tomar decisiones o implementar estrategias para generar un efecto deseado en el fenómeno de interés [2].

La actividad económica de la compañía Accenture se basa, entre otras cosas, en proveer servicios en estrategia y consultoría donde se desarrollan herramientas para los clientes a partir de datos que ellos suministran, y con base en las cuales se logra extraer información adicional relevante, ya sea para fortalecer el conocimiento del negocio y tomar mejores decisiones o para generar estrategias orientadas al aumento o disminución de alguna variable de interés [3]. Como parte de este proceso, es necesario garantizar que las herramientas desarrolladas generen

resultados confiables y legibles, basados en estrategias y herramientas en el marco de lo que se conoce como *Business Intelligence* [4].

Uno de los casos de uso en los que se trabajó consistía en un sistema de optimización, basado en modelos de analítica avanzada, donde se buscaba obtener la mejor combinación entre precio por unidad de los productos y descuentos, de modo que se maximizara el margen bruto (diferencia entre las ventas y los costos) de 6 SKU* priorizados en regiones y canales específicos, para una empresa productora de alimentos. Este sistema constaba de varias etapas entre las que se encuentra la ingesta de datos, el procesamiento de los datos, desarrollo de modelos, inferencia y despliegue [5].

En este caso de uso se implementaron dos tipos de modelos analíticos: Modelos *ElasticNet* [6] que incorporan penalización tipo *Lasso* y *Ridge*, y modelos de regresión lineal múltiple [7], los cuales presentaban algunas inconsistencias en los resultados entregados para 2 de los productos priorizados. Asimismo, el tablero de visualización o *dashboard* utilizado en el despliegue de los resultados, el cual fue desarrollado en el *framework* de desarrollo web *Dash*, no contaba con la claridad y comprensibilidad deseada, pues las características gráficas de los elementos que la componen no permitían interpretar de forma fácil los resultados. Es por esto por lo que se requería hacer un diagnóstico detallado en el que utilizando el método Delphi [8] se construyera una lista de los elementos que debían ser mejorados, considerando la relación costo-beneficio y la opinión de un grupo de expertos tanto del grupo de analítica como de parte del cliente, para luego implementar dichos cambios requeridos y así garantizar la satisfacción del cliente con el producto entregado.

3. Objetivos

3.1. Objetivo General

Implementar mejoras en los modelos predictivos y la herramienta de despliegue de resultados en sistema de optimización de margen bruto, realizando un diagnóstico mediante el método Delphi, con el fin de garantizar la funcionalidad, reducir el tiempo de ejecución y facilitar la interacción del usuario con la herramienta.

* *Stock-Keeping Unit*. Código único para cada producto.

3.2. Objetivos Específicos

- Diagnosticar el sistema de optimización de margen bruto realizando un estudio detallado de sus componentes, tomando como referencia el método Delphi, donde se identifiquen oportunidades de mejoramiento, tanto de funcionalidad y eficiencia, como de interpretabilidad de los resultados, obteniendo un inventario de los elementos a mejorar y la manera en la que se validarán.
- Aplicar las modificaciones identificadas en el diagnóstico asociadas a la funcionalidad y eficiencia, ajustando los modelos predictivos y los algoritmos usados, para garantizar la validez de los resultados, la disminución del tiempo de ejecución y la fluidez de la visualización.
- Diseñar e implementar una nueva versión del dashboard del sistema haciendo uso del framework para el desarrollo de aplicaciones web Dash, considerando las necesidades del cliente y los elementos determinados en el diagnóstico, con el objetivo de desplegar en la nube el sistema mejorado en su interpretabilidad y visualización de los resultados.
- Validar las mejoras implementadas en el sistema de optimización siguiendo las métricas definidas en el diagnóstico.

4. Marco Teórico

En el desarrollo de este proyecto se hace necesario el manejo de algunos conceptos asociados a las diferentes áreas del conocimiento involucradas, desde definiciones hasta conceptos técnicos relacionados con las herramientas a utilizar, las cuales se tratan a continuación.

4.1 Método Delphi

El método Delphi hacer referencia a una serie de procesos grupales estructurados, para sondear la opinión de un grupo de expertos y llegar a una respuesta común, donde las opiniones, creencias y juicios individuales se recopilan y organizan de manera asistemática [8]. El funcionamiento de este método se basa en la elaboración de un cuestionario que debe contestar cada experto. Una vez analizados los resultados globales, se vuelve a realizar otro cuestionario que deberán contestar los mismos expertos, tras darles a conocer los resultados obtenidos en la consulta anterior. El proceso puede repetirse varias veces hasta alcanzar cierto nivel de consenso. Finalmente, el responsable del estudio elaborará sus conclusiones a partir de la explotación estadística de los datos obtenidos [9]. El método Delphi se basa en:

- El anonimato de los participantes

- La repetitividad y la retroalimentación controlada
- La respuesta del grupo en forma estadística.

4.2. Análisis de datos

El análisis de datos es un proceso que consiste en inspeccionar, limpiar y transformar datos con el objetivo de resaltar información útil, para sugerir conclusiones y apoyo en la toma de decisiones [10]. Esta ciencia puede descomponerse en 3 tipos de análisis: Análisis descriptivo, predictivo y prescriptivo [11].

- **Análisis descriptivo:** El análisis descriptivo se utiliza cuando la organización tiene un gran conjunto de datos sobre eventos pasados o sucesos históricos. Para utilizar estos datos se debe realizar un proceso de limpieza, ordenamiento, transformación y visualización con el fin de que sean entendibles para la audiencia a la que se quieren comunicar. En este tipo de análisis es común observar tableros de control, gráficas de barras, gráficas de pasteles, infografías y otros.
- **Análisis predictivo:** El análisis predictivo se refiere a la aplicación de técnicas y modelos matemáticos o estadísticos a una gran cantidad de datos con el objetivo de poder pronosticar con cierta probabilidad lo que podría llegar a suceder. Con este tipo de análisis se logra entender las correlaciones entre variables y cómo podrían comportarse en un futuro. En este contexto, el *Machine Learning* que consiste en darle la habilidad de aprender a un sistema de cómputo, permite con cierto grado de confiabilidad pronosticar un resultado.
- **Análisis prescriptivo:** El análisis prescriptivo, le recomienda rutas de acción que puede seguir una empresa. Además, cuantifica el efecto de cada una de estas acciones para ayudar a tomar las mejores decisiones en pro de buscar los objetivos de negocio de la organización, como por ejemplo incursionar en un nuevo mercado, ubicar un producto en áreas específicas de un almacén con mejores probabilidades de venta o mitigar un riesgo que pueda enfrentar.

4.3. Regresión ElasticNet

Una regresión lineal es un método estadístico que trata de modelar la relación entre una variable continua y una o más variables independientes mediante el ajuste de una ecuación lineal [12]. Este tipo de regresión lineal normaliza el vector de coeficientes, lo que permite generar un modelo en el que solo algunos de los coeficientes sean no nulos. Su función de coste es equivalente a [13]:

$$RSS_{elastic\ net} = \sum_{i=1}^n (y_i - f(x_i))^2 + \alpha \left(\lambda \sum_{j=1}^p \beta_j^2 + (1 - \lambda) \sum_{i=0}^n |\beta_j| \right)$$

Este tipo de modelo fuerza los coeficientes del modelo a que tiendan a cero, minimizando así el riesgo de overfitting, reduciendo varianza, atenuado el efecto de la correlación entre predictores y reduciendo la influencia en el modelo de los predictores menos relevantes.

4.4. Regresión Lineal Múltiple

Una regresión lineal es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente, variables independientes y un término aleatorio. La regresión lineal múltiple permite generar un modelo lineal en el que el valor de la variable dependiente (**Y**) se determina a partir de un conjunto de variables independientes llamadas predictores (**X1, X2, X3...**) [14]. Los modelos de regresión múltiple pueden emplearse para predecir el valor de la variable dependiente o para evaluar la influencia que tienen los predictores sobre ella y siguen la siguiente ecuación:

$$Y_i = (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_n X_{ni}) + e_i$$

β_0 : Es el valor de la variable Y cuando todos los predictores son cero.

β_i : Es el efecto promedio que tiene el incremento en una unidad de la variable predictora X_i sobre la variable dependiente Y.

e_i : Es el residuo o error determinado por la diferencia entre el valor observado y el estimado por el modelo.

4.5. Framework de desarrollo de aplicaciones web Dash

Dash es un *framework* de código abierto para Python que se utiliza en la creación de aplicaciones web analíticas. Es una poderosa biblioteca que simplifica el desarrollo de aplicaciones basadas en datos, especialmente útil para los científicos de datos de Python que no están muy familiarizados con el desarrollo web, permitiéndoles crear *dashboards* en su navegador con una gran cantidad de herramientas disponibles [15]. Este *framework* está construido sobre *Plotly.js*, *React* y *Flask*, vinculando elementos modernos de la interfaz de usuario como menús desplegables, controles deslizantes, y todo tipo de gráficos (barras, de dispersión, diagramas de cajas y bigotes, histogramas, etc.) construidos a partir de los modelos analíticos o resultados generados desde Python.

A través de patrones simples, *Dash* abstrae todas las tecnologías y protocolos que se requieren para crear una aplicación web *full-stack* con visualización de datos interactiva, donde mediante la metodología de *callbacks* [16] se generan funciones que *Dash* llama automáticamente cada vez que cambia la propiedad de un componente de entrada, con el objetivo de actualizar alguna propiedad en otro componente.

4.6. Servicios de Amazon AWS

Amazon Web Services (AWS) es una plataforma en la nube muy completa y ampliamente adoptada del mundo, que ofrece más de 200 servicios completos de centros de datos en todo el mundo. Millones de clientes, incluidas las empresas emergentes de más rápido crecimiento, las empresas más grandes y las principales agencias gubernamentales, utilizan AWS para reducir costos, volverse más ágiles e innovar más rápido [17]. Los servicios web AWS permiten llevar a cabo el flujo completo de trabajo, desde la ingesta de datos, preprocesamiento hasta el despliegue. Entre los servicios utilizados en este proyecto se encuentran:

- **Amazon S3:** Servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en la industria [18].
- **Amazon SageMaker:** Este servicio ayuda a los científicos y desarrolladores de datos a preparar, crear, entrenar e implementar rápidamente modelos de aprendizaje automático de alta calidad al reunir un amplio conjunto de capacidades especialmente diseñadas para el aprendizaje automático [19].
- **Amazon Elastic Beanstalk:** Es un servicio fácil de usar para implementar y escalar aplicaciones y servicios web desarrollados con *Java*, *.NET*, *PHP*, *Node.js*, *Python*, *Ruby*, *Go* y *Docker* en servidores familiares como *Apache*, *Nginx*, *Passenger* y *IIS* [20].

5. Metodología

A continuación, se describe cada uno de los pasos realizados y técnicas empleadas para el desarrollo del proyecto.

5.1. Diagnóstico del sistema de optimización de margen bruto mediante el método Delphi.

El sistema de optimización de margen bruto contaba con una serie de oportunidades de mejora, por lo que inicialmente se realizó un entendimiento de dicho sistema, con el objetivo de conocerlo a profundidad y tener así la facultad de proponer y aportar elementos para su mejoramiento.

5.1.1 Aplicación del método Delphi

Como parte del diagnóstico se aplicó el método Delphi para generar la ruta a seguir en el mejoramiento del sistema de optimización. El primer paso consistió en determinar los elementos que debían ser analizado, los cuales fueron:

- Validez de los resultados de los modelos.
- Secciones que debería incluir el dashboard.
- Ubicación de los elementos visuales.
- Tipo de gráfico para presentar la información histórica de las variables costo, ventas, margen bruto, precio de lista (pdl) y precio de venta al público (pvp).
- Tipo de gráfico para presentar el precio de los competidores.
- Tipo de gráfico para presentar la atribución a las ventas de diferentes variables.
- Tipo de gráfico para presentar los KPI's asociados a las variables de unidades de producto, ventas, costos, margen bruto, descuentos, precios.

Con base en estos elementos, y para lograr aplicar el método Delphi que consiste en una serie de cuestionarios controlados para recolectar y entregar información buscando lograr un consenso grupal, se determinaron una serie de preguntas acerca de la opinión de los participantes sobre estos ítems, cuestionando la validez de los resultados del modelo y estableciendo un conjunto de formatos de visualización de datos apropiados considerando la teoría reportada en la literatura y los formatos de visualización de datos que eran fáciles de implementar usando la librería Plotly.

5.1.2 Elementos del sistema a mejorar

Una vez recopilada la opinión de los expertos del equipo de analítica y el equipo de los clientes, se establecieron los siguientes puntos como elementos de mejora del sistema:

- Realizar nuevamente el entrenamiento de dos de los SKU's priorizados debido a la falta de coherencia con los resultados obtenidos.
- Modificar el dashboard de manera que se tengan sólo dos pestañas, incluyendo una sección de información histórica y otra sección de simulación donde se generen escenarios y se observen los resultados de la optimización.
- Representar los KPI's asociados a las variables de unidades de producto, ventas, costos, margen bruto, descuentos, precios, por medio de diagramas de barras donde se incluya el valor actual, el valor optimizado y el valor simulado.

5.2. Ajuste de algoritmos usados y modelos predictivos.

En el diagnóstico realizado se logró identificar una inconsistencia en los resultados obtenidos para 2 de los modelos predictivos y una serie de afectaciones en la eficiencia desde el punto de vista algorítmico, por lo que en esta sección se describen los ajustes y modificaciones realizados en este sentido.

5.2.1 Actualización de los modelos para 2 de los SKU's incluidos

El margen bruto se define como el beneficio directo que obtiene una empresa por un bien o servicio. En este contexto este modelo optimizador usa todas las variables que se consideraron como drivers de las ventas para hallar la combinación óptima de precio (PdL – PVP) y descuentos (Descuento Total) al cuál se debe vender cada SKU y así obtener el mayor margen bruto posible para la empresa productora de alimentos.

$$\text{Margen Bruto} = \text{PdL} \times \text{Ventas}(\text{variables}) - f(\text{descuentos}) - f(\text{costo total de producción})$$

En este caso, las ventas están modeladas haciendo uso de una regresión lineal múltiple acompañada de la estrategia de regularización Elastic Net, donde la elección adecuada de las variables tiene una relación directa con el desempeño del modelo. En dos de los SKUs priorizados se identificaron algunas variables que influían de manera negativa en el desempeño del modelo, ya que no permitían explicar el valor de las ventas, así que se procedió a reemplazar estas variables, y se llevó a cabo nuevamente el entrenamiento de estos modelos, obteniendo una mejora en la métrica de desempeño que se eligió.

5.2.2 Modificación de los algoritmos y mejora en la eficiencia del código

En la generación de los diferentes gráficos y elementos visuales se detectaron algunas redundancias que disminuían la eficiencia en la generación de la información. Se realizó entonces un análisis de los diferentes módulos de código que componían el sistema, identificando una alta complejidad computacional que no permitía la generación eficiente de los visuales, así que se buscó la manera de sintetizar este código, para reducir la complejidad computacional y mejorar el tiempo de cargue de la información en el dashboard.

5.3. Implementación de una nueva versión del dashboard usando el framework Dash.

Otro de los elementos principales donde se identificaron algunas oportunidades de mejora fue el dashboard en el que se presentan los resultados del sistema. Este dashboard estaba compuesto inicialmente por 3 pestañas:

- **Pestaña de histórico:** Visualización de información histórica de algunas variables de interés como precio de lista, precio de venta al público, costos, etc.
- **Pestaña de optimización:** Visualización de los resultados del modelo que optimiza el margen bruto.
- **Pestaña de simulación:** Panel donde se generan posibles escenarios de acuerdo con los valores de las variables ingresados.

5.3.1 Modificación de la cantidad de pestañas del dashboard

Con el fin de contar con un dashboard más compacto se redujo la cantidad de pestañas con las que se contaba, pasando de 3 pestañas a 2. En la Figura 1¹ se visualiza el panel superior con las 3 pestañas que se tenían anteriormente, y en la Figura 2 las pestañas disponibles después de las modificaciones.

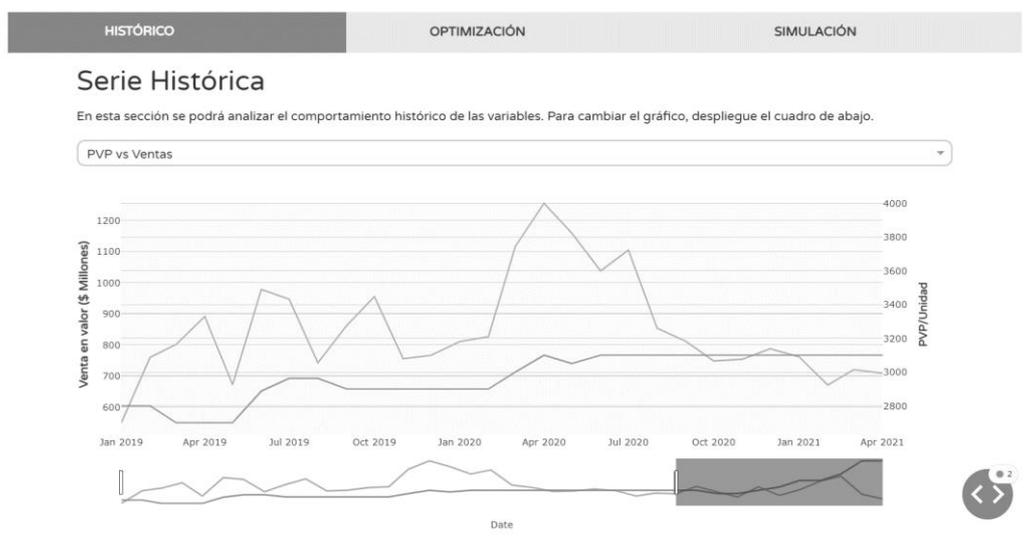


Figura 1. Pestañas disponibles del dashboard antes de las modificaciones.

¹ Las figuras se presentan en escala de grises para conservar oculta la identidad de la empresa productora de alimentos.

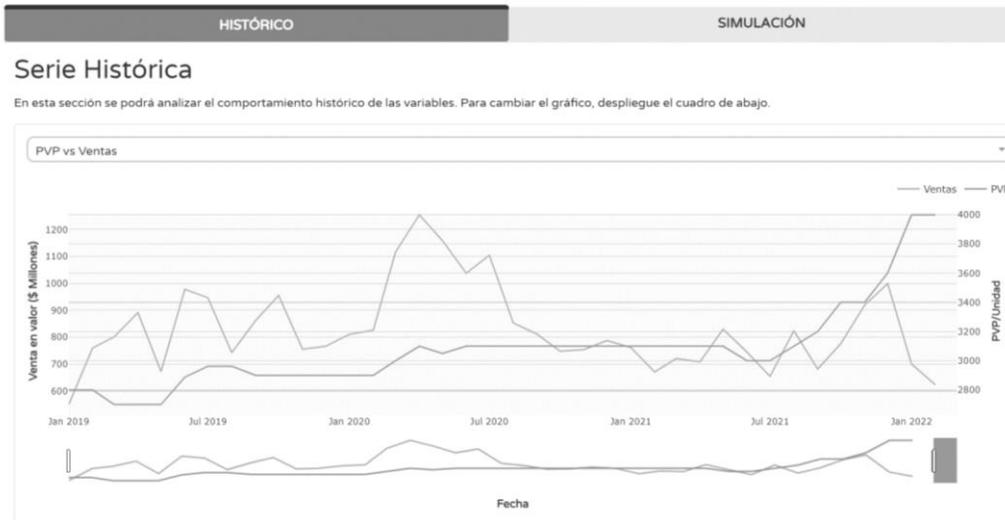


Figura 2. Pestañas disponibles en el dashboard después de las modificaciones.

5.3.2 Modificación de sección de optimización y simulación

El dashboard contaba con una sección de optimización y otra sección de simulación, pero estas dos pestañas se consolidaron en una sola pestaña que se denominó simulación. En esta sección se podía encontrar diversos gráficos asociados a variables de interés como los precios de los competidores, datos históricos, variables que más aportan a las ventas, KPI's y los resultados tanto de la optimización como de la simulación.

Se llevó a cabo una redistribución inicialmente de la posición de estos gráficos, ofreciendo una vista más compacta y consolidada. En la Figura 3 se muestra el estado de esta pestaña antes de la simulación y en la Figura 4 se ilustra el resultado después de las modificaciones.

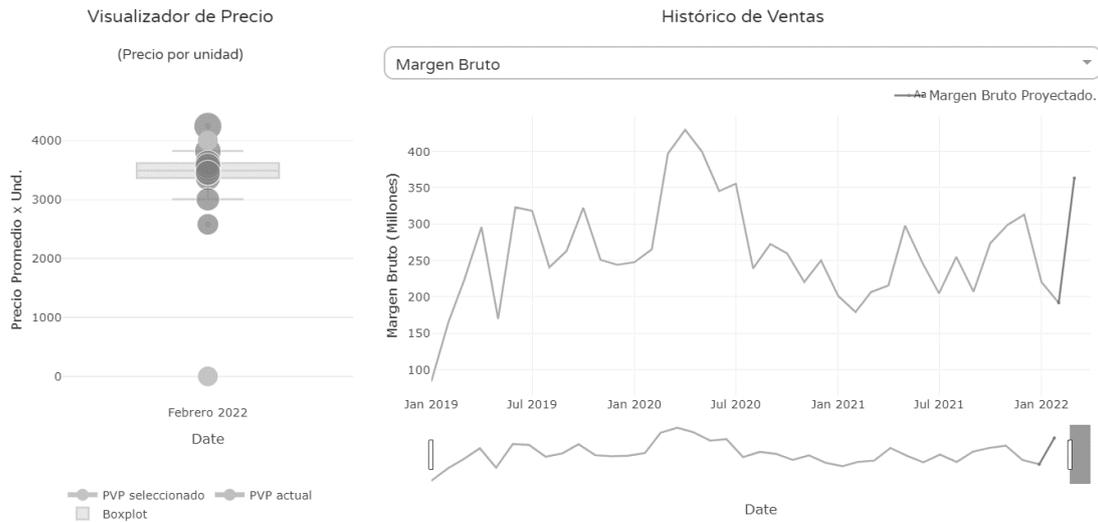


Figura 3. Disposición de los elementos gráficos en la pestaña de simulación antes de las modificaciones

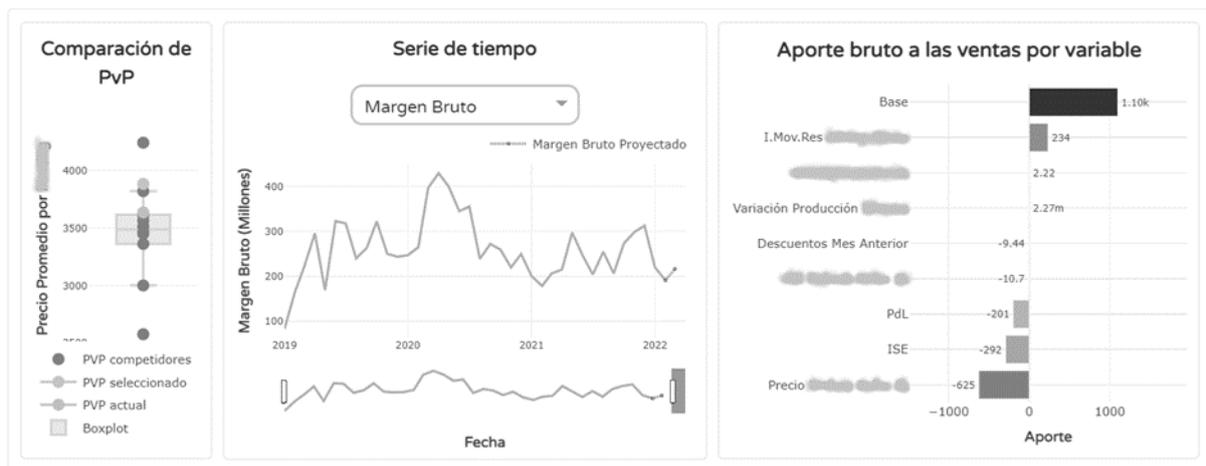


Figura 4. Disposición de los elementos gráficos en la pestaña de simulación después de las modificaciones realizadas

Otro elemento importante de esta sección era los KPI's asociados a las variables cantidad de unidades, ventas, margen bruto, descuentos y precios. En la Figura 5 puede verse la forma en la que inicialmente se presentaban estos indicadores, la cual generaba una alta carga cognitiva, por lo que se reemplazaron por un conjunto de gráficos de barras incluyendo los valores actuales, optimizados y simulados (Figura 6).



Figura 5. Visualización de los KPI correspondientes a las variables de interés antes de las modificaciones

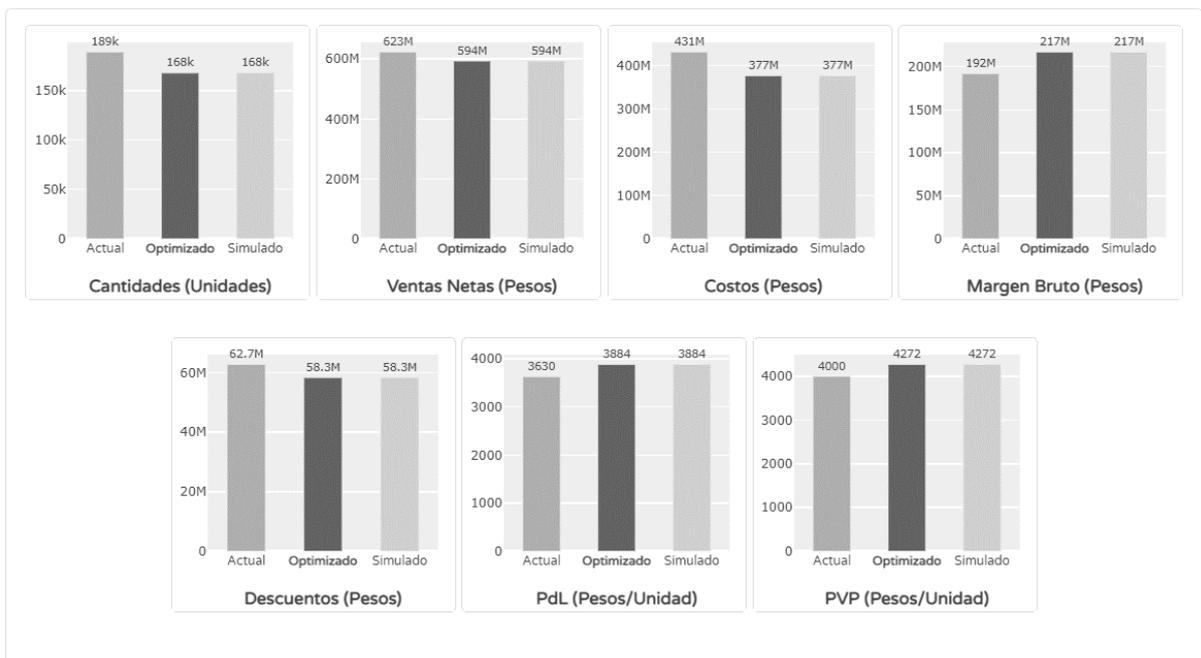
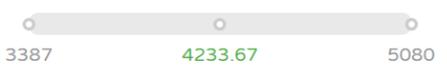


Figura 6. Visualización de los KPIs después de las modificaciones.

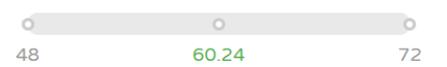
Por último, existía un panel de simulación donde le usuario podía ingresar dos variables que permitan generar un escenario en el que se calculaba un valor de margen bruto de acuerdo con dicha información ingresada. Este panel se observa en la Figura 7.

Ingrese un precio de prueba (PVP)



3387 4233.67 5080

Ingrese un descuento de prueba.



48 60.24 72

*Se están utilizando valores iniciales/predeterminados.

* Ingrese un número que figure dentro de ambos rangos para realizar una simulación. Para volver a los valores predeterminados, ingrese 0 EN AMBOS CAMPOS o ingrese el valor inicial señalado en verde.

Figura 7. Panel de simulación antes de las modificaciones.

Teniendo en cuenta el objetivo de sintetizar el tablero, este panel se ajustó en un contenedor vertical más angosto y el resultado se muestra en la Figura 8.

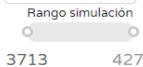
Simulación

Información [?](#)

Ingrese un PVP (Pesos)

\$

Rango simulación

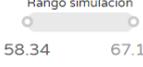


3713 4272

Ingrese un descuento (Millones)

M\$

Rango simulación



58.34 67.12

Figura 8. Panel de simulación después de las modificaciones

5.3.3 Creación de ventana de autenticación

La información desplegada en el dashboard es de carácter confidencial, de modo que a la hora de desplegar el dashboard en internet cualquier persona con el enlace podría acceder a este tablero. Por esto se desarrolló un sistema básico de autenticación con el que se haría control de las personas con acceso permitido.

Este sistema de autenticación consistió en un diccionario creado desde Python donde se generaron pares usuario-contraseña con la información de las personas a las que se les permitiría el acceso al dashboard (Figura 9).

```
Userbase64Enconding.py
Userbase64Enconding.py > ...
1 import json
2 import base64
3
4 USERNAME_PASSWORD_PAIRS = {
5     'usuario1': '123456789'
6     , 'usuario2': '123456789'
7 }
8
```

Figura 9. Diccionario con los pares usuario-contraseña que tendrían acceso al dashboard

Estos pares usuario-contraseña se codificaron usando una codificación en base 64, la cual se efectúa de la siguiente manera:

- Se convierten los caracteres del texto que se va a codificar en su equivalente ASCII (a: 97, b: 98, etc).
- Los valores decimales obtenidos en la etapa anterior se convierten a sus equivalentes binarios (por ejemplo, 97: 01100001).
- Todas las tramas binarias se concatenan, obteniendo una trama binaria de mayor longitud.
- Esta trama binaria resultante se divide en secciones iguales de 6 bits cada una.
- Los conjuntos iguales de 6 bits se convierten en sus equivalentes decimales.
- Por último, los equivalentes decimales se convierten en sus valores Base 64 (es decir, 4: E).

Con este procedimiento se logró obtener una cadena de caracteres que representa las credenciales codificadas. Esta cadena de caracteres corresponde a una variable de entorno que se usa para el despliegue, la cual se explica con mayor detalle en una sección posterior.

Por otro lado, se desarrolló una interfaz de usuario usando la librería Plotly, donde los usuarios podían introducir sus credenciales para ser validadas (Figura 10²).

² El logo de la compañía se ocultó para que sea anónimo



Figura 10. Interfaz del sistema de autenticación del dashboard

Luego de que el usuario ingresara las credenciales en estos campos de la interfaz de ingreso, se realizó un proceso de decodificación, que corresponde al proceso inverso donde se toma la variable de entorno que es la cadena de caracteres codificada, y se decodifica para obtener el diccionario de vuelta y validar si los datos ingresados por el usuario están dentro del diccionario.

5.3.4 Modificación del tutorial del dashboard

En el dashboard estaba incluido un manual de usuario para entender las diferentes vistas, gráficas y procedimientos. Este manual era una única pieza gráfica la cual se reemplazó por una presentación con la técnica de carrusel, donde se diseñó un tutorial completo para facilitar la usabilidad del tablero (Figura 11).



Figura 11. Implementación del manual de usuario con la técnica carrusel.

5.3.5 Despliegue en el servicio AWS Elastic Beanstalk

El dashboard donde se muestran los resultados de los modelos se modificó y actualizó de forma local, pero se buscaba que los clientes pudieran visualizar y hacer uso de dicho tablero de forma remota. Para esto se hizo uso de las capacidades de los servicios en la nube, los cuales permiten desplegar una aplicación web de manera sencilla, evitando preocupaciones con respecto a la arquitectura.

En el diagrama que se muestra en la Figura 12 se puede observar la arquitectura que se definió para este caso de uso, donde se utilizan diferentes servicios de AWS los cuales interactúan



Figura 12. Arquitectura del sistema de optimización de margen bruto utilizando los servicios

Existían diversas alternativas (Servicios) para desplegar aplicaciones web usando AWS, pero se optó por una de las más sencillas que es el uso del servicio AWS Elastic Beanstalk, el cual es un servicio para implementar y escalar aplicaciones y servicios web. Con este servicio cloud se cargó nuestro código y este servicio se encargó de manejar automáticamente la implementación, desde la asignación de capacidad, el balanceo de carga y el escalado automático hasta el monitoreo del estado de la aplicación [21].

6. Resultados y análisis

En esta sección se describen los resultados obtenidos y el análisis de las mejoras implementadas en el sistema optimizador de margen bruto.

6.1 Análisis de las métricas de desempeño de los modelos

Para determinar el desempeño y la precisión de los modelos, existen varias medidas del error. El error se refiere a qué tan cerca está el pronóstico generado por el modelo de lo que ocurre realmente.

El criterio bajo el cual se evaluaron los modelos es el Error Porcentual Absoluto Medio (Mean Absolute Percentage Error o MAPE). El hecho que se estime una medida de error porcentual (%) hace más fácil su interpretación por lo que suele ser frecuentemente utilizado al realizar pronósticos. La ecuación para calcular el MAPE es:

$$MAPE = \frac{\sum_{t=1}^n \frac{|A_t - F_t|}{|A_t|}}{n}$$

Donde A_t representa el valor real de la variable dependiente (valor real de la venta), F_t es el pronóstico para el valor de esa variable calculado por el modelo de regresión, y n indica el número de observaciones que se tienen.

Para el primer SKU al que se le modificaron las variables con las que se entrenaba el modelo se obtuvo el siguiente valor de MAPE:

```
MAE Train: 7.91MM -- MAE Test: 4.83MM
MAPE Train: 0.09 -- MAPE Test: 0.07
```

Se obtuvo un valor de MAPE de test del 7%, lo que indica que el modelo obtenido con las variables introducidas como predictores es capaz de explicar el 93% de la variabilidad observada en las ventas.

Para el segundo SKU al que se le modificaron las variables con las que se entrenaba el modelo se obtuvo el siguiente valor de MAPE:

```
El error rmse de test es: 0.20212906735497396
El error (mape) de test es: 0.07235487111026775
```

Se obtuvo un valor de MAPE de test del 7.2%, lo que indica que el modelo obtenido con las variables introducidas como predictores es capaz de explicar el 92.8% de la variabilidad observada en las ventas.

6.2 Análisis del tiempo de respuesta en los elementos del dashboard

Gracias a las mejoras en la parte algorítmica, considerando las afectaciones de la eficiencia en la generación de los elementos visuales que se tenían, se logró disminuir considerablemente el tiempo de cargue de la información en el dashboard, y en la Tabla 1 se puede visualizar la comparación de los tiempos que se tarda en visualizar

la información en el tablero antes y después de las medicaciones, así como el porcentaje de reducción en dicho tiempo.

Tabla 1. Comparación de los tiempos de cargue de la información antes y después de las modificaciones realizadas.

Elemento del dashboard	Tiempo cargue información antes de las modificaciones	Tiempo cargue información antes de las modificaciones	Porcentaje reducción del tiempo
Gráfico con información histórica	6.4 s	4.3 s	32.81 %
Boxplot con datos de competidores	3.8 s	3.6 s	5.26 %
Gráfico de atribución de las variables a las ventas	5.9 s	4.1s	30.51 %
KPIs con variables relevantes.	5.4 s	4.7 s	12.96 %
Actualización de los gráficos después de simular	6.9 s	4.9 s	28.99 %

7. Conclusiones

El diagnóstico realizado sobre el sistema de optimización de margen bruto hizo posible la implementación de los ajustes o modificaciones necesarias para garantizar la satisfacción del cliente. Las actualizaciones y mejoras realizadas en el sistema optimizador de margen bruto permitieron garantizar la coherencia de los resultados entregados, aumentar la eficiencia al momento de cargar la información, brindar seguridad de la información desplegada gracias al sistema de autenticación y obtener escalabilidad en cuanto a la cantidad de personas con acceso al tablero. Asimismo, el sistema optimizador de margen bruto pudo ser entregado al cliente, garantizando los requerimientos acordados.

8. Referencias bibliográficas.

- [1] "Big Data Industries: 5 Industries Being Reshaped by Data Analytics." <https://www.precisely.com/blog/big-data/big-data-industries-data-analytics> (accessed Feb. 24, 2022).
- [2] S. Brandt and C. Methods, *Data Analysis*. .
- [3] D. Fisher, R. DeLine, M. Czerwinski, and S. Drucker, "Interactions with big data analytics," *Interactions*, vol. 19, no. 3, pp. 50–59, May 2012, doi:

10.1145/2168931.2168943.

- [4] S. Negash and P. Gray, "Business Intelligence," *Handb. Decis. Support Syst.* 2, pp. 175–193, 2008, doi: 10.1007/978-3-540-48716-6_9.
- [5] "5 Steps of Data Analysis | Analytics Steps." <https://www.analyticssteps.com/blogs/5-steps-data-analysis> (accessed Feb. 25, 2022).
- [6] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stat. Soc. Ser. B (Statistical Methodol.)*, vol. 67, no. 2, pp. 301–320, Apr. 2005, doi: 10.1111/J.1467-9868.2005.00503.X.
- [7] M. Tranmer, J. Murphy, M. Elliot, and M. Pampaka, "Multiple Linear Regression (2 nd Edition)," 2020, Accessed: Feb. 25, 2022. [Online]. Available: <https://hummedia.manchester.ac.uk/institutes/cmist/a>.
- [8] J. Steurer, "The Delphi method: An efficient procedure to generate knowledge," *Skeletal Radiol.*, vol. 40, no. 8, pp. 959–961, 2011, doi: 10.1007/s00256-011-1145-z.
- [9] C. E. George Reyes and L. Trujillo Liñan, "Aplicación del Método Delphi Modificado para la Validación de un Cuestionario de Incorporación de las TIC en la Práctica Docente," *Rev. Iberoam. Evaluación Educ.*, vol. 11, no. 1, pp. 113–135, 2018, doi: 10.15366/riee2018.11.1.007.
- [10] "Data analysis : a model-comparison approach : Judd, Charles M : Free Download, Borrow, and Streaming : Internet Archive." <https://archive.org/details/dataanalysismode0000judd/page/n663/mode/2up> (accessed Feb. 27, 2022).
- [11] "3 tipos de análisis de datos para mejorar la toma de decisiones." <https://www.pragma.com.co/blog/3-tipos-de-analisis-de-datos-para-mejorar-la-toma-de-decisiones> (accessed Feb. 27, 2022).
- [12] J. A. Rodrigo, "Regularización Ridge, Lasso y Elastic Net con Python," 2020. <https://www.cienciadedatos.net/documentos/py14-ridge-lasso-elastic-net-python.html>.
- [13] "Elastic Net | Interactive Chaos." <https://interactivechaos.com/en/node/884> (accessed Feb. 28, 2022).
- [14] "Introducción a la Regresión Lineal Múltiple." https://www.cienciadedatos.net/documentos/25_regresion_lineal_multiple (accessed Feb. 28, 2022).
- [15] "Dash Framework." https://www.tutorialspoint.com/python_web_development_libraries/python_web_development_libraries_dash_framework.htm (accessed Feb. 28, 2022).
- [16] "Part 3. Basic Callbacks | Dash for Python Documentation | Plotly."

- <https://dash.plotly.com/basic-callbacks> (accessed Feb. 28, 2022).
- [17] "What is AWS." <https://aws.amazon.com/what-is-aws/> (accessed Feb. 28, 2022).
- [18] "What is Amazon S3? - Amazon Simple Storage Service." <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html> (accessed Feb. 28, 2022).
- [19] "Amazon SageMaker - Machine Learning Solution." https://aws.amazon.com/pm/sagemaker/?trk=ps_a134p000007BxfHAAS&trkCampaign=acq_paid_search_brand&sc_channel=PS&sc_campaign=acquisition_LATAMO&sc_publisher=Google&sc_category=Machine Learning&sc_country=LATAMO&sc_geo=LATAM&sc_outcome=acq&sc_detail=amazon sagemaker&sc_content=Sagemaker_e&sc_matchtype=e&sc_segment=532500830539&sc_medium=ACQ-P%7CPS-GO%7CBrand%7CDesktop%7CSU%7CMachine Learning%7CSagemaker%7CLATAMO%7CEN%7CText&s_kwcid=AL!4422!3!532500830539!e!!g!!amazon sagemaker&ef_id=CjwKCAiAgvKQBhBbEiwAaPQw3EEuKW82KXnniMD_aNbel5Oef1_EitUCHblsWyypUhhj3ciG_jTskBoCWhkQAvD_BwE:G:s&s_kwcid=AL!4422!3!532500830539!e!!g!!amazon sagemaker (accessed Feb. 28, 2022).
- [20] "AWS Elastic Beanstalk – Deploy Web Applications." <https://aws.amazon.com/elasticbeanstalk/> (accessed Feb. 28, 2022).
- [21] T. Hava, "What is AWS Elastic Beanstalk?," 2021. <https://www.hava.io/blog/what-is-aws-elastic-beanstalk>.
-