



# **Manual de usuario Modelo fijación de precios para maximización de rentabilidad Frubana**

**Elaborado por:**  
Cristhian Amaya / Andrés Beltrán / Rodrigo Méndez





## Contenido

<b>Introducción</b> .....	3
<b>Acceso a los datos</b> .....	3
Acceso a información exógena .....	3
<b>Preparación de datos</b> .....	4
<b>Modelado</b> .....	4
Modelado de regresión .....	5
Modelo de optimización.....	6
<b>Tablero de control</b> .....	6
<b>Paso a paso de despliegue</b> .....	7
<b>Repositorio</b> .....	8
<b>Recomendaciones</b> .....	9
<b>Anexos</b> .....	9

## Introducción

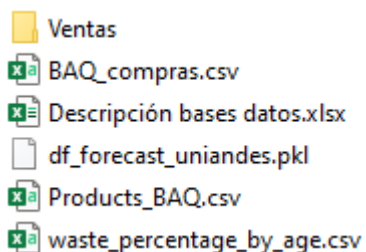
El manual de usuario tiene como objetivo ser una guía para la ejecución del modelo de fijación de precios de manera periódica.

En este documento encontrara el listado de los insumos necesarios para una correcta ejecución del modelo de principio a fin, el cual, se construyó pensado para ser ejecutado de manera local utilizando diferentes herramientas (Cuaderno de Jupyter notebook, Python) para obtener los resultados de los modelos desarrollados. Adicional, el modelo se ejecuta por fases comenzando con el acceso a los datos suministrados por frubana para el desarrollo del proyecto, preparación de datos para el modelado en donde se importaran las fuentes de datos y serán alistadas para obtener las tablas con las que se construirán los modelos, modelado incluyendo modelos de regresión que identifican las relaciones entre cantidad vendida de un producto con los precios y cantidades de otros productos, modelos de optimización para determinar los precios recomendados que maximizarían las utilidades, se describe el despliegue de un tablero de control construido para visualizar los resultados de los modelos generados, adicional, se encuentra un apartado llamado Paso a paso de despliegue el cual, especificará el proceso de ejecución de principio a fin, material necesario para la ejecución por parte de un usuario que desee implementarlo, seguido se presenta una descripción de repositorio del proyecto en GitHub, finalmente, se presentan algunas recomendaciones generales para tener en cuenta en dicha ejecución.

## Acceso a los datos

Fuentes de datos suministradas por frubana para generación de modelos se listan a continuación:

La carpeta contiene 3 archivos csv, 1 archivo de xlsx, y un archivo en.pkl. La carpeta de ventas contiene 12 archivos, uno por mes en un formato .pkl.



Estas fuentes de datos las encuentra en el siguiente link del repositorio

[https://github.com/cramayag/Pricing\\_Frubana/tree/main/Datos%20Frubana/Datos%20Frubana](https://github.com/cramayag/Pricing_Frubana/tree/main/Datos%20Frubana/Datos%20Frubana)

## Acceso a información exógena

Se realiza la descarga de los archivos SIPSA de forma Manual del sitio web del DANE (<https://www.dane.gov.co/index.php/estadisticas-por-tema/agropecuario/sistema-de-informacion-de-precios-sipsa/mayoristas-boletin-semanal-1/boletin-mayorista-semanal-2024>) donde se descarga la actualización que se desea realizar y se adiciona en la carpeta 'Datos Externos/input/SIPSA Semanal', también es necesario ajustar en el archivo Datos Externos/input/fechas.xlsx, el nombre del archivo y el periodo con la fecha que comprende.

## Preparación de datos

Para la preparación de datos se tiene dos cuadernos los cuales son:

### Cuadernos de la preparación de datos

- Analisis exploratorio.ipynb

[https://github.com/cramayag/Pricing\\_Frubana/blob/main/Datos%20Frubana/Analisis%20exploratorio.ipynb](https://github.com/cramayag/Pricing_Frubana/blob/main/Datos%20Frubana/Analisis%20exploratorio.ipynb)


El cual tiene como resultado un archivo .csv con el nombre de **base consolidada.csv** ubicado en el siguiente link:


[https://github.com/cramayag/Pricing\\_Frubana/tree/main/Datos%20Externos/Interim](https://github.com/cramayag/Pricing_Frubana/tree/main/Datos%20Externos/Interim)


- Pipeline\_externo.ipynb

Este cuaderno se ejecuta del archivo unificar\_semanas\_sipsa.py que crea tres archivos de Excel (data1, data2, data3), cada archivo contiene información de productos consolidada semanalmente del SIPSA, según la clasificación de tres tablas categorizadas según la tipología de los productos (1.Verduras y hortalizas;2.Frutas frescas;3.Tubérculos, raíces y plátanos) con precios de SIPSA para generar correlaciones entre cantidades y precios, y toma la base\_consolidada generada del cuaderno análisis exploratorio el cual prepara la información generando las fuentes de datos que se usaran en la fase de modelado. ([https://github.com/cramayag/Pricing\\_Frubana/tree/main/Datos%20Externos/code](https://github.com/cramayag/Pricing_Frubana/tree/main/Datos%20Externos/code))

### Salida de la preparación de datos

 base\_de\_datos.csv

 correlaciones\_cantidad.csv

 correlaciones\_precios.csv

[https://github.com/cramayag/Pricing\\_Frubana/tree/main/Modelos%20Pricing/Input](https://github.com/cramayag/Pricing_Frubana/tree/main/Modelos%20Pricing/Input)


## Modelado

En esta fase, una vez preparados los datos, se construye un modelo por cada producto estos modelos fueron elegidos bajo diversas pruebas con diferentes modelos eligiendo aquel que presentara menor error por producto, en esta fase se construyó un cuaderno con pruebas el cual lo encuentra en el siguiente link


[https://github.com/cramayag/Pricing\\_Frubana/tree/main/Modelos%20Pricing/Model](https://github.com/cramayag/Pricing_Frubana/tree/main/Modelos%20Pricing/Model)

### Modelado de regresión


Con las fuentes de datos generadas en el punto anterior

 `base_de_datos.csv`


---

 `correlaciones_cantidad.csv`


---

 `correlaciones_precios.csv`


Se construye un modelo por cada producto (5 productos)

 `Empaquetamiento modelo ajo.ipynb`


---

 `Empaquetamiento modelo cebolla cabezona.ipynb`


---

 `Empaquetamiento modelo name.ipynb`

---

 `Empaquetamiento modelo platano verde.ipynb`


---

 `Empaquetamiento modelo tomate chonto.ipynb`


[https://github.com/cramayag/Pricing\\_Frubana/tree/main/Modelos%20Pricing](https://github.com/cramayag/Pricing_Frubana/tree/main/Modelos%20Pricing)

Cada cuaderno construye un modelo y lo empaqueta para ser usado en la fase de despliegue cada modelo genera un archivo con los parámetros de los modelos en un formato .pkl


Salidas de la ejecución de los cuadernos

 `ajo.pkl`


---

 `cebolla_cabezona.pkl`


---

 `name.pkl`

---

 `platano_verde.pkl`

---

 `tomate_chonto.pkl`

[https://github.com/cramayag/Pricing\\_Frubana/tree/main/Modelo%20de%20Optimizacion/Input](https://github.com/cramayag/Pricing_Frubana/tree/main/Modelo%20de%20Optimizacion/Input)

## Modelo de optimización

En la carpeta de 'Modelo de Optimización' encontraras el archivo. ipynb con nombre 'Modelo de Optimización'. Este archivo se alimenta de los modelos predictivos construidos en la fase anterior, y csv llamada 'base\_de\_datos.csv', ubicada en la carpeta de Input de los modelos de Pricing. Al correr todo el notebook, la aplicación pasara a construir una heurística para seleccionar los parámetros (precios de los productos de cada predictor), de manera que se obtenga el set de precios que maximizan la contribución por producto con base en los modelos predictivos construidos en la fase anterior.

```
def optimization(sku, sep, model, t=0):
    #Generamos el set de combinaciones de precios que vamos a evaluar en los modelos predictivos
    entry_np = generate_data(sku, sep)
    #Costo base de la variable top 5
    cost_i = list(get_cost(get_variables(sku)).values())[0]

    df = pd.DataFrame()
    cantidad = []
    contribucion = []

    #Si t = 0, entonces el modelo es un modelo predictivo para OLS con constante en el modelo
    if t == 0:
        for i in entry_np:
            x = np.hstack([np.ones((i.shape[0], 1)), i])
            q = model.predict(x)
            cantidad.append(q[0])
            contribucion.append(q[0]*(i[0][0]-cost_i))
    #Si t = 1, entonces el modelo es un modelo predictivo para GAM, en este caso solo sería para el tomate
    elif t == 1:
        for i in entry_np:
            q = model.predict(i)
            cantidad.append(q[0])
            contribucion.append(q[0]*(i[0][0]-cost_i))
    #Si t = 2, entonces el modelo es un modelo predictivo OLS pero con transformación exp para interpretar los resultados log
    elif t == 2:
        for i in entry_np:
            q = model.predict(i)
            q_i = np.exp(q[0])
            cantidad.append(q_i)
            contribucion.append(q_i*(i[0][0]-cost_i))

    df['parametros'] = entry_np
    df['cantidad'] = cantidad
    df['contribucion'] = contribucion

    return df[df.contribucion==max(df.contribucion)]
```

La tabla resultante es una tabla con los precios sugeridos para cada producto top 5, así como el precio que se sugiere para los otros productos que sirvieron para predecir los 5 productos con mayor cantidad de ventas.

mapa\_estados.csv

resultados.csv

## Tablero de control

El tablero de control se desarrolla bajo el Framework de Python utilizado para crear aplicaciones web interactivas, el tablero de control cuenta con las siguientes componentes, donde:

assets	Agregando el dashboard final
components	Agregando el dashboard final
data	Agregando el dashboard final
pages	Agregando el dashboard final
application.py	Agregando el dashboard final
requirements.txt	Agregando el dashboard final

- assets: contiene el archivo styles el cual le da la forma, colores al dashboard
- components: carpeta base que podría contener código para desarrollar gráficos como mapas o graficas interactivas en el dashboard en un desarrollo futuro.
- data: bases de datos de los archivos principales.
- pages: Contiene los archivos .py uno para cada una de las paginas que tiene el aplicativo.
- application.py: contiene el archivo base de desarrollo de la aplicación

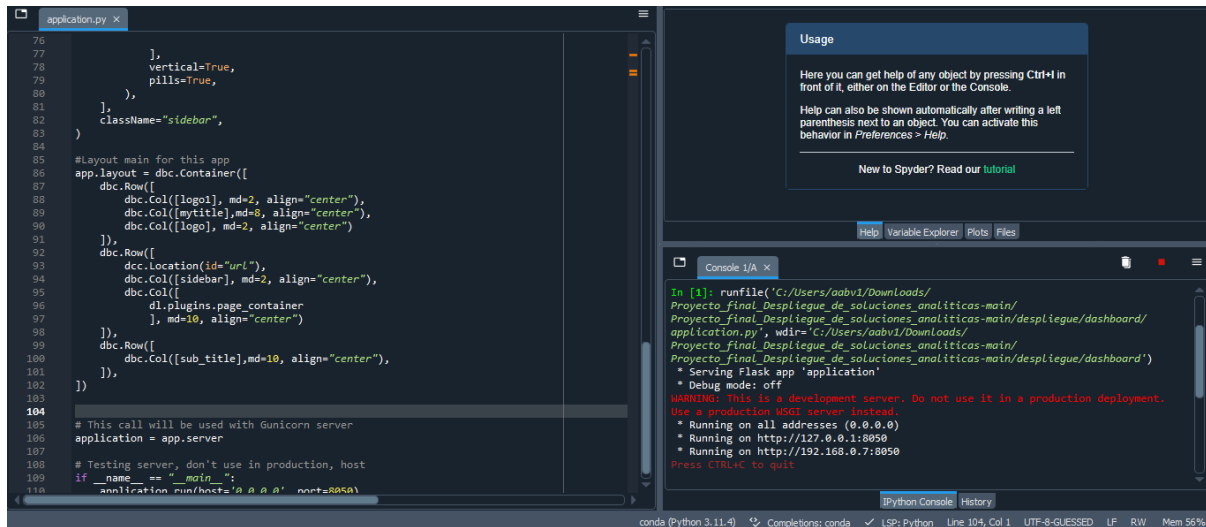
Además de instalar los paquetes utilizados en las diferentes etapas del proyecto, es necesario tener las siguientes librerías dash-bootstrap-components y dash.

## Paso a paso de despliegue

Para realizar el despliegue, se realiza de forma manual, en primer lugar, hay que realizar una clonación del repositorio ([https://github.com/cramayag/Pricing\\_Frubana/tree/main](https://github.com/cramayag/Pricing_Frubana/tree/main)), se puede ejecutar directamente el dashboard para observar el funcionamiento de los datos predichos por el modelo. Si se desea realizar una actualización con datos diferentes a los utilizados se puede guiar de las consideraciones anteriormente descritas para realizar el respectivo análisis

Para ejecutar el tablero, dentro de la carpeta Dashboard se encuentra el archivo: 'application.py', el cual se puede abrir por medio de una terminal o un entorno de desarrollo como Spyder o Visual Studio. Hay que asegurarse de tener las dependencias necesarias de librerías dash-bootstrap-components y dash.

Al ejecutar la aplicación se genera el acceso por medio del enlace <http://ip:8050>, como en el ejemplo de la imagen <http://192.168.0.7:8050>.



```
76
77     ],
78     vertical=True,
79     pills=True,
80   ),
81   ],
82   className="sidebar",
83 )
84
85 #Layout main for this app
86 app.layout = dbc.Container([
87     dbc.Row([
88         dbc.Col([logo], md=2, align="center"),
89         dbc.Col([mytitle], md=8, align="center"),
90         dbc.Col([logo], md=2, align="center")
91     ]),
92     dbc.Row([
93         dbc.Col([location(id="url"),
94                 dbc.Col([sidebar], md=2, align="center"),
95                 dbc.Col([
96                     dl.plugins.page_container
97                     ], md=10, align="center")
98                 ], md=10, align="center")
99     ]),
100     dbc.Col([sub_title], md=10, align="center"),
101 ]),
102
103
104
105 # This call will be used with Gunicorn server
106 application = app.server
107
108 # Testing server, don't use in production, host
109 if __name__ == "__main__":
110     application.run(host='0.0.0.0', port=8050)
```

Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Help Variable Explorer Plots Files

Console I/A x

```
In [1]: runfile('C:/Users/aabv1/Downloads/
Proyecto_final_Despliegue_de_soluciones_analiticas-main/
Proyecto_final_Despliegue_de_soluciones_analiticas-main/despliegue/dashboard/
application.py', wdir='C:/Users/aabv1/Downloads/
Proyecto_final_Despliegue_de_soluciones_analiticas-main/
Proyecto_final_Despliegue_de_soluciones_analiticas-main/despliegue/dashboard')
* Serving Flask app 'application'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:8050
* Running on http://192.168.0.7:8050
Press CTRL+C to quit
```

Python Console History

conda (Python 3.11.4) Completions: conda LSP: Python Line 104, Col 1 UTF-8-GUESSED LF RW Mem 56%

Se puede cargar pegando la dirección dada por la consola en el navegador de preferencia. Al ejecutar se podrá interactuar con el tablero de mando, donde en el costado izquierdo comprende el menú de páginas contenidas.



En la hoja Resultados Pricing observarás los resultados de los modelos para dar la recomendación de precios de los productos.

## Repositorio





Dashboard
Datos Externos
Datos Frubana
Modelos Pricing
Primera entrega
.gitattributes
README.md

[https://github.com/cramayag/Pricing\\_Frubana/tree/main](https://github.com/cramayag/Pricing_Frubana/tree/main)

## Recomendaciones

El modelo prescriptivo presentado con estos modelos es un modelo desarrollado para ofrecer sugerencias de precios de algunos productos específicos. Si se quiere desarrollar otros modelos específicos, es necesario contar con un mayor volumen de histórico de ventas de los productos a los que se desea modelar.

La sugerencia de fijación de precios debe ser evaluada constantemente. El modelo tendrá predicciones más precisas a medida que se le entrene con más datos de volúmenes de ventas.

Al depender de los precios de otros productos, puede ser necesario en el futuro que la compañía evalúe categorizar productos por importancia, de manera que los productos que tengan menor importancia sean considerados en la construcción de modelos de productos con mayor relevancia. La relevancia de cada producto está sujeta a la consideración del departamento de mercadeo, y puede depender de su cantidad, utilidad, nivel de ventas, estrategia de portafolio, etc.

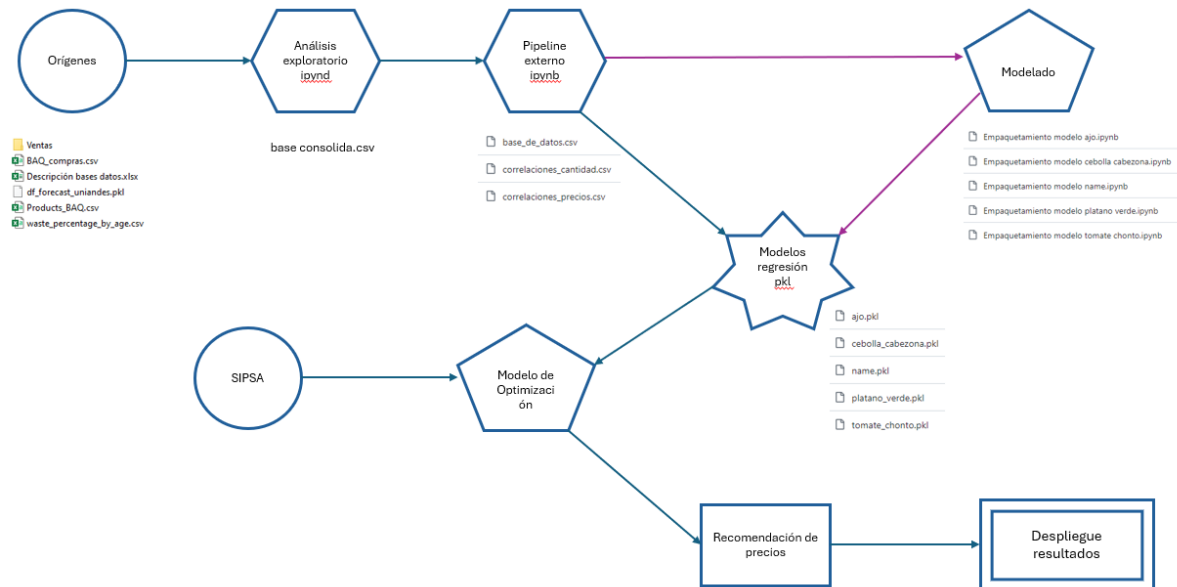
En este modelo no se consideraron los impactos de las políticas de descuentos, por lo que su implementación debe ser independiente a las estrategias de descuentos para aumentar volumen.

Es importante contar con personas que tengan conocimiento de modelos de regresión lineal y optimización para el mantenimiento y la apropiación de esta solución.

## Anexos

- i. Diagrama esquemático propuesto.


Se describe el proceso de ejecución




El flujo para el despliegue periódico se debe hacer siguiendo las conexiones azules.

- ii. Reporte técnico de experimentos  
En el siguiente link de Git Hub encontrará el reporte técnico de experimentos, junto con el notebook donde se desarrollaron.  
[https://github.com/cramayag/Pricing\\_Frubana/tree/main/Entregas%20de%20reportes/Segunda%20Entrega](https://github.com/cramayag/Pricing_Frubana/tree/main/Entregas%20de%20reportes/Segunda%20Entrega)
- iii. Rubrica de evaluación  
En el siguiente link de Git Hub encontrarán la rubrica de evaluación.  
[https://github.com/cramayag/Pricing\\_Frubana/blob/main/Entregas%20de%20reportes/Tercera%20Entrega/Rubrica%20de%20pruebas.xlsx](https://github.com/cramayag/Pricing_Frubana/blob/main/Entregas%20de%20reportes/Tercera%20Entrega/Rubrica%20de%20pruebas.xlsx)
- iv. Archivos de código desarrollados




 Dashboard


---

 Datos Externos


---

 Datos Frubana


---

 Modelos Pricing


---

 Primera entrega

---

 .gitattributes

---

 README.md

[https://github.com/cramayag/Pricing\\_Frubana/tree/main](https://github.com/cramayag/Pricing_Frubana/tree/main)