**Objetivo inicial:** Predecir el valor percibido por los clientes que será medido por el precio, usando diferentes características del producto.

**Nuevo Objetivo:** Encontrar la cantidad de ventas que se pueden realizar dada una propuesta de valor, con el fin de encontrar los precios más bajos a los que se puede vender conservando ganancias operativas.

Nos replanteamos el objetivo porque las correlaciones o influencia del precio en las demás variables era casi nula. Además, habíamos estado usando variables dependientes cómo entradas del modelo y estas no son controlables por el mercante.

**Iteraciones para mejorar:**

1. Tener un modelo de clasificación en el que asociemos una palabra a un grupo de palabras similares y proyectar las ventas que se pueden hacer si se añade esta palabra como tag al producto.

**Características a evaluar inicialmente:**

\*Las características de las imágenes.

\*Calificación del mercante.

\*Cantidad de productos.

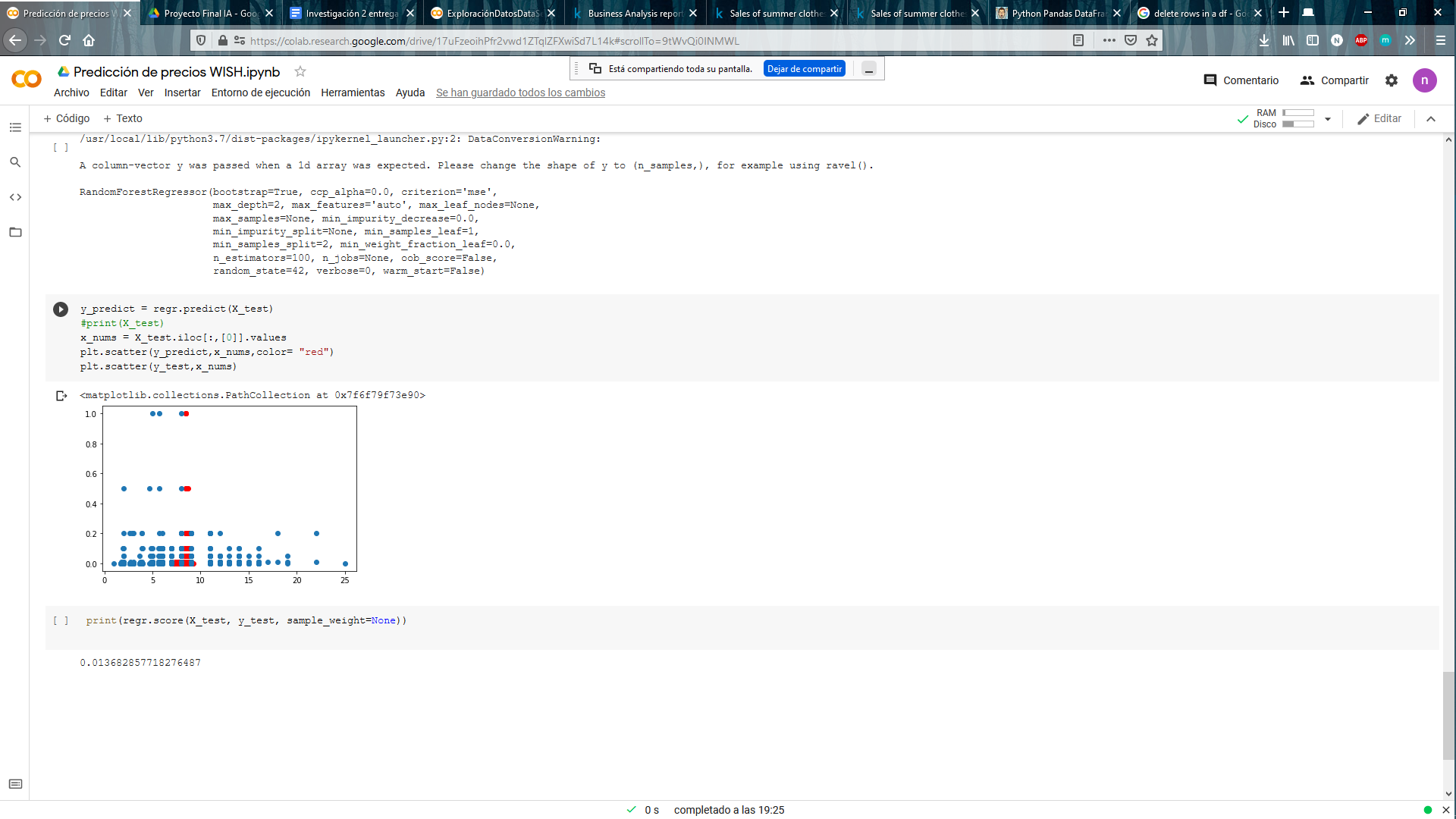
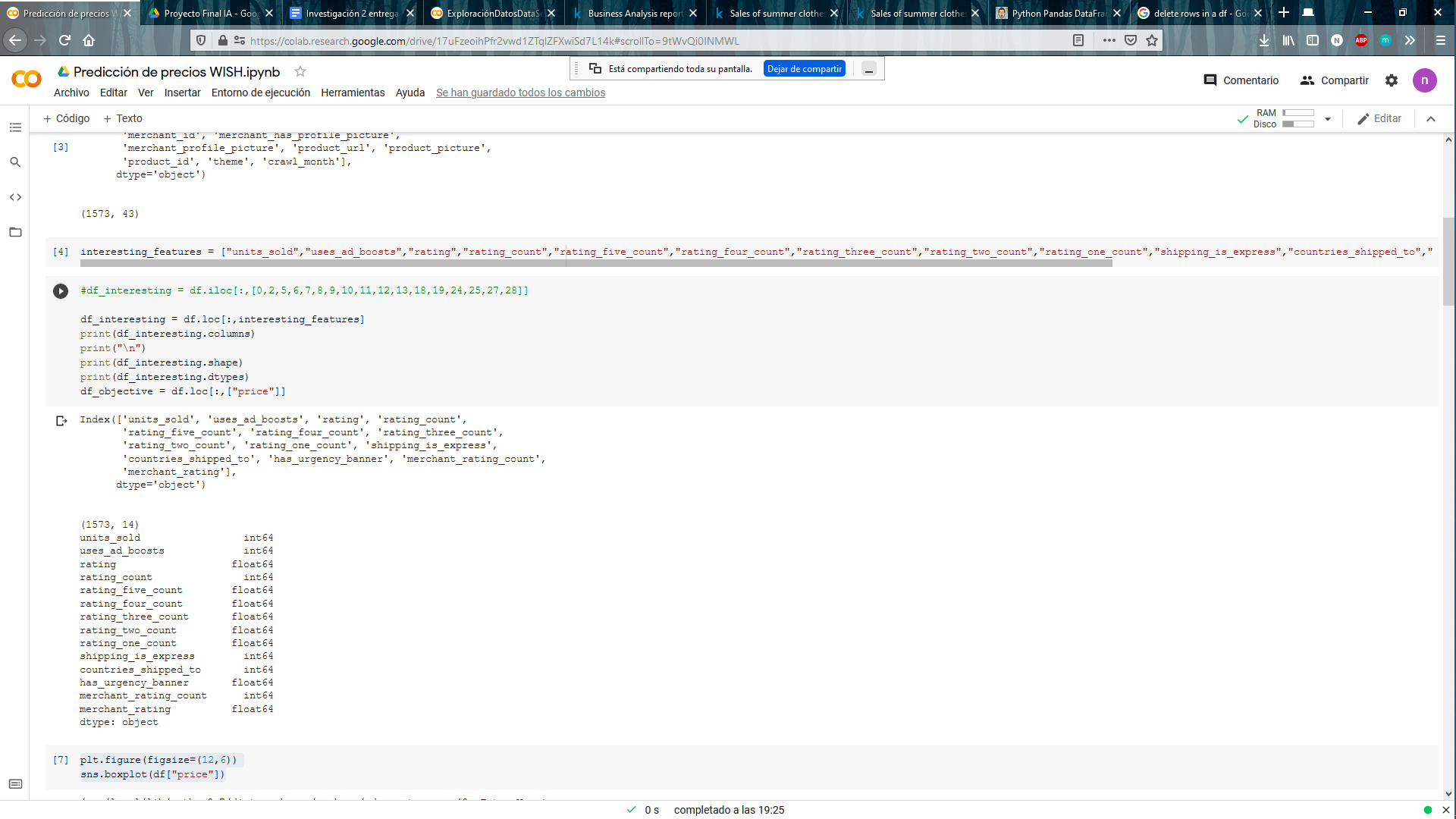
\*Calificación del producto.

**Características evaluadas al final:**

["price","units\_sold","uses\_ad\_boosts","rating","rating\_count","merchant\_rating\_count","merchant\_rating"]

Modelo 1: Predicciones con RandomForestRegressor

Para generar un modelo de línea base simplemente cogimos todos los valores numéricos y tratamos de hacer un modelo de regresión con ellos. Los resultados definitivamente fueron malos con un R2 = 0,013



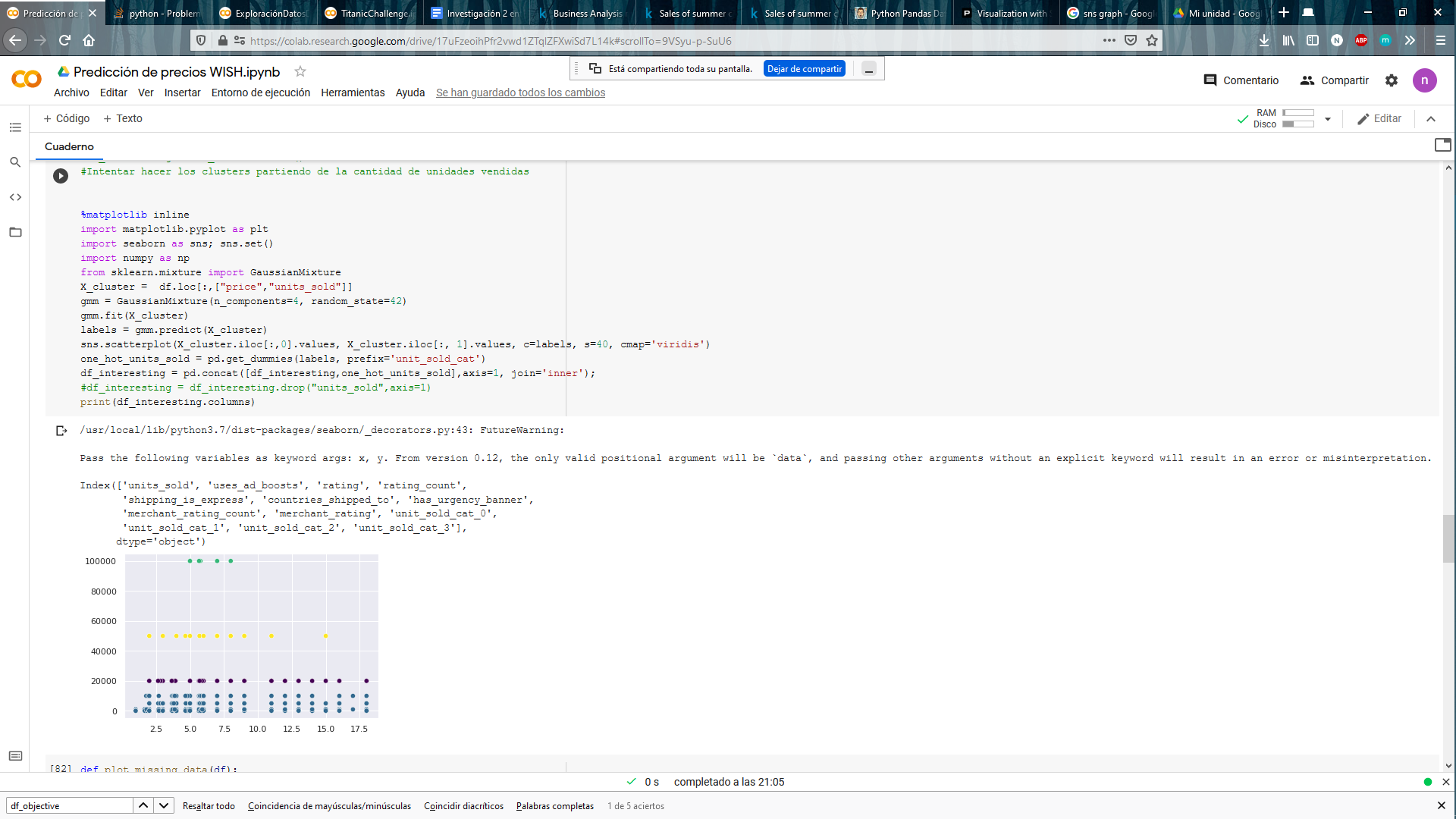
Modelo 2: Predicciones con RandomForestRegressor

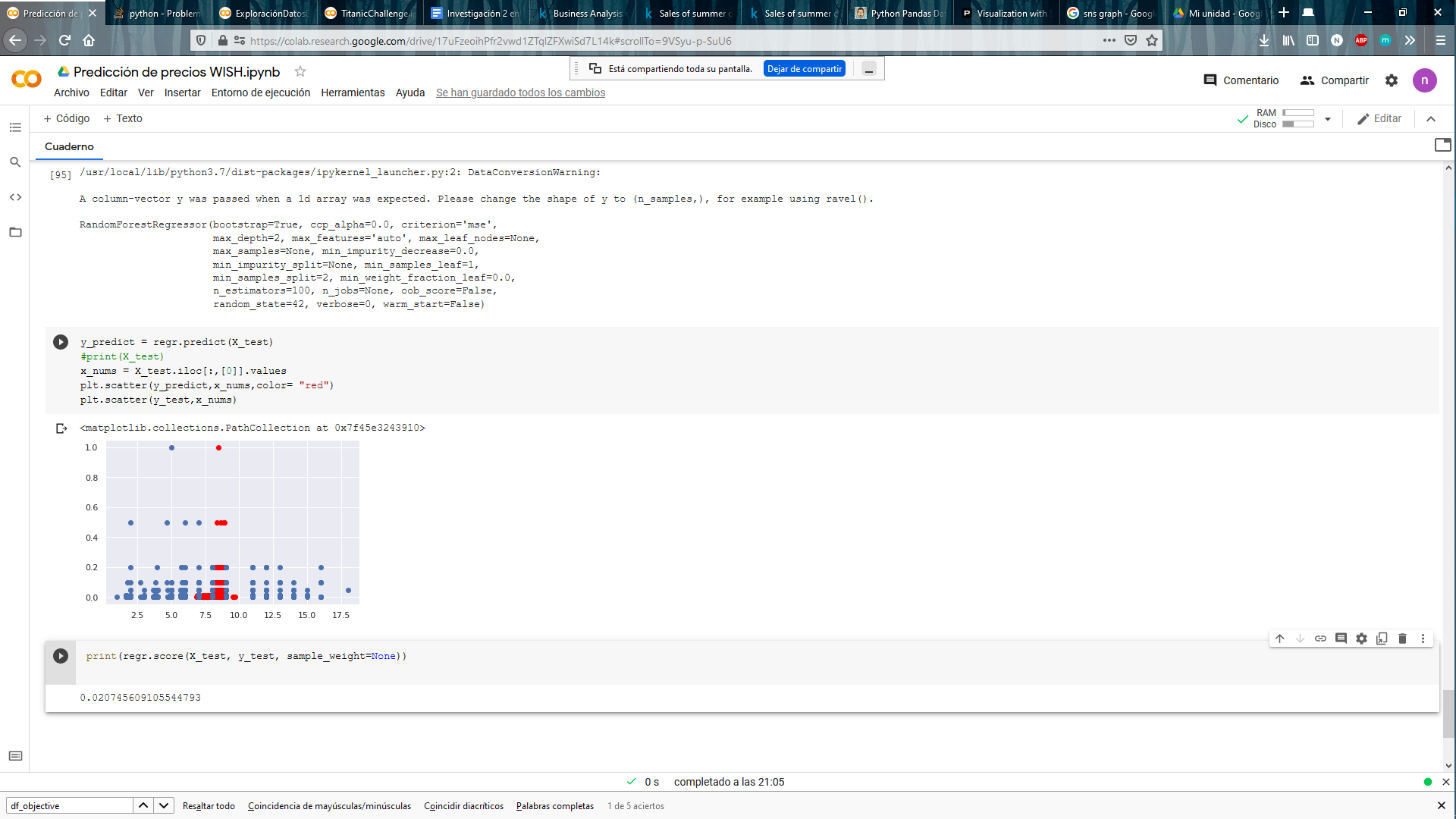
Después de los resultados del primer experimento empezamos a hacer hipótesis y experimentos sobre las variables que debíamos considerar más importantes.

Al inicio queríamos predecir el precio y nos pareció importante eliminar los outliers del precio.

Creamos nuevas gráficas para observar relaciones entre las variables, y con dichas gráficas decidimos eliminar los rating\_x\_count porque aportan la misma información que el rating count.

Con las nuevas gráficas vimos una estructura interesante entre el precio y la cantidad vendida, y decidimos hacer un experimento donde creamos una nueva categoría con los clusters de units sold y el precio.





Modelo 3: Predicciones con RandomForestRegressor

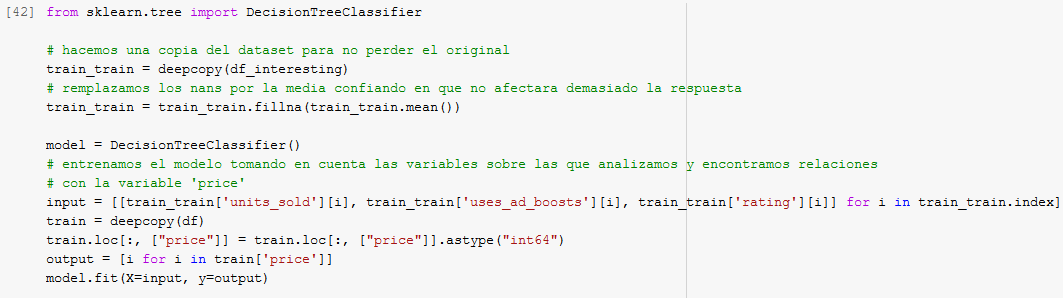
Combinar Rating-Rating Count

Lista de atributos interesantes: interesting\_features = ["price","units\_sold","uses\_ad\_boosts","rating","rating\_count","shipping\_is\_express","countries\_shipped\_to","merchant\_rating\_count","merchant\_rating"]

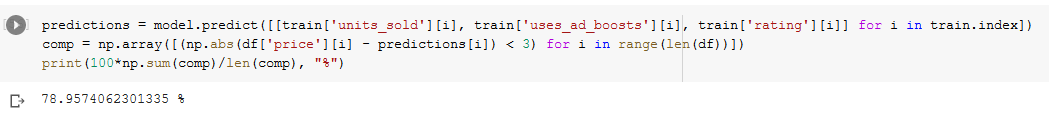


Modelo 4: Predicciones con DesicionTreeClassifier

1. Convertir el precio a int64 para poder usar un modelo de predicción discreta.
2. Entrenar el modelo con las variables qué se identificaron cómo significativas en el análisis (units\_sold, rating y uses\_ad\_boosts)



1. Probamos la efectividad del modelo con un margen de error de +-3



Modelo 5: Predicciones con DesicionTreeClassifier:

1. Convertir el precio a int64 para poder usar un modelo de predicción discreta.
2. Entrenar el modelo con las variables que se identificaron cómo significativas en el análisis: "units\_sold","uses\_ad\_boosts","rating","rating\_count","shipping\_is\_express","countries\_shipped\_to","merchant\_rating\_count","merchant\_rating"



Modelo 6: Predicciones con support vector machine:

1. Convertir el precio a int64 para poder usar un modelo de predicción discreta.
2. Entrenar el modelo con las variables que se identificaron cómo significativas en el análisis: "units\_sold","uses\_ad\_boosts","rating","rating\_count","shipping\_is\_express","countries\_shipped\_to","merchant\_rating\_count","merchant\_rating"

