Proyecto III: Análisis Coffee shop

1. Objetivo

El objetivo de este proyecto es responder a cuatro preguntas, ***¿Cómo optimizar el beneficio de las ventas de una cafetería?, ¿Cuáles son las horas en las que más se vende? , según el día y la hora, ¿Habrá alguna venta? y según la edad, el género, la hora y el día, ¿Qué producto es más probable que se venda?***

2. Extracción de datos

Estos datos los vamos a obtener de un Dataset preestablecido, de la página "Kaggle"; https://www.kaggle.com/datasets/ylchang/coffee-shop-sample-data-1113?select=201904+sales+reciepts.csv

3. Preparación de datos

i. Limpieza de datos

Tenemos un dataset que simula la venta de un "Coffee shop". Realizamos un filtrado de datos, eliminamos los valores nulos o incoherentes y suprimimos las columnas innecesarias para este proyecto.

ii.LabelEncoder

Hacemos uso de herramientas como "LabelEncoder" para codificar variables categóricas.

4. Exploración de datos

i. Gráficos Simples

En esta sección, ya tenemos los datos filtrados y procesados, podemos empezar a visualizarlos con gráficos simples, como un "hist2d" o "scatter", ambas de la librería matplotlib.

ii. Exploración no gráfica

Eventualmente podremos hacer uso de funciones predeterminadas como “describe()” de modo a poder realizar una exploración no-gráfica del dataset, además de customizar con el comando “style.background_gradient” con el objetivo de hacernos una idea más intuitiva y general del conjunto de datos.

5. Modelización

i. Modelos y variables

Con los datos preparados, podemos iniciar el proceso de modelización. Dado nuestro dataset, elegimos el “target” para la variable a predecir y los “features” como aprendizaje para el modelo. Recalcar, que este proyecto lo abordaremos desde un punto de vista “Supervisado”, es decir, proporcionamos al modelo toda la información necesaria.

ii. Ejecución

Dada la naturaleza de nuestro problema, usaremos modelos de clasificación. Elegiremos un set de varios tipos (KMeans, KNeighborsClassifier, RandomForestClassifier, etc…) y se tendrá en cuenta los que mejores resultados arrojen.

iii. Diagnóstico y comparación

Tras ejecutar los modelos y seleccionar el más óptimo, podremos hacer uso de funciones como “cross_val_score” para saber cuál es la mejor manera de particionar nuestro set “train” y “test” o bien “GridSearchCV” para encontrar los mejores hiperparámetros que se ajustan al dataset de estudio.

Además podremos hacer uso de la matriz de confusión para observar, de más cerca, el rendimiento de nuestro modelo.

5. Presentación

i. Presentación

El modelo ya se ha elegido, mejorado y preformado. Podemos mostrar los resultados en una tabla o gráfico. Lo importante será comparar los datos predichos con los reales.

Resumen del Proyecto

Nuestro proyecto es un "Coffee Shop", se compone de varios datasets separados, reuniendo información sobre las ventas de productos, género y edad de los clientes, etc..

transaction_date	transaction_time	customer_id	product_id	quantity	unit_price
2019-04-12	06:50:31	0	48	1	2.50
2019-04-15	08:29:48	0	8	1	45.00
2019-04-15	08:29:48	0	36	1	3.75
2019-04-15	10:42:05	0	69	1	3.25
2019-04-15	10:42:05	0	35	1	3.10

customer_id	gender	birth_year
1	M	1950
2	M	1950
3	M	1950
4	M	1950
5	M	1951

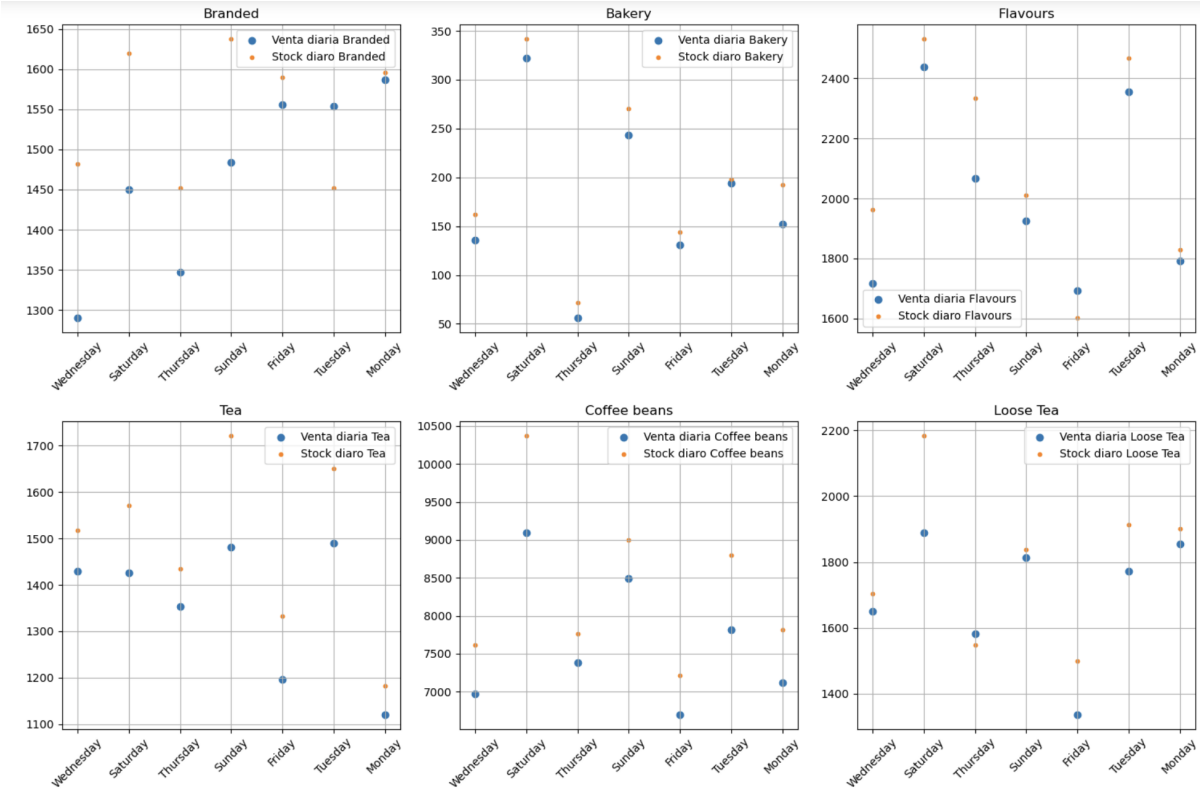
product_id	waste	quantity_sold
69	10	8
70	6	12
71	10	8
72	39	9
73	9	9

product_id	product_category
1	Coffee beans
2	Coffee beans
3	Coffee beans
4	Coffee beans
5	Coffee beans

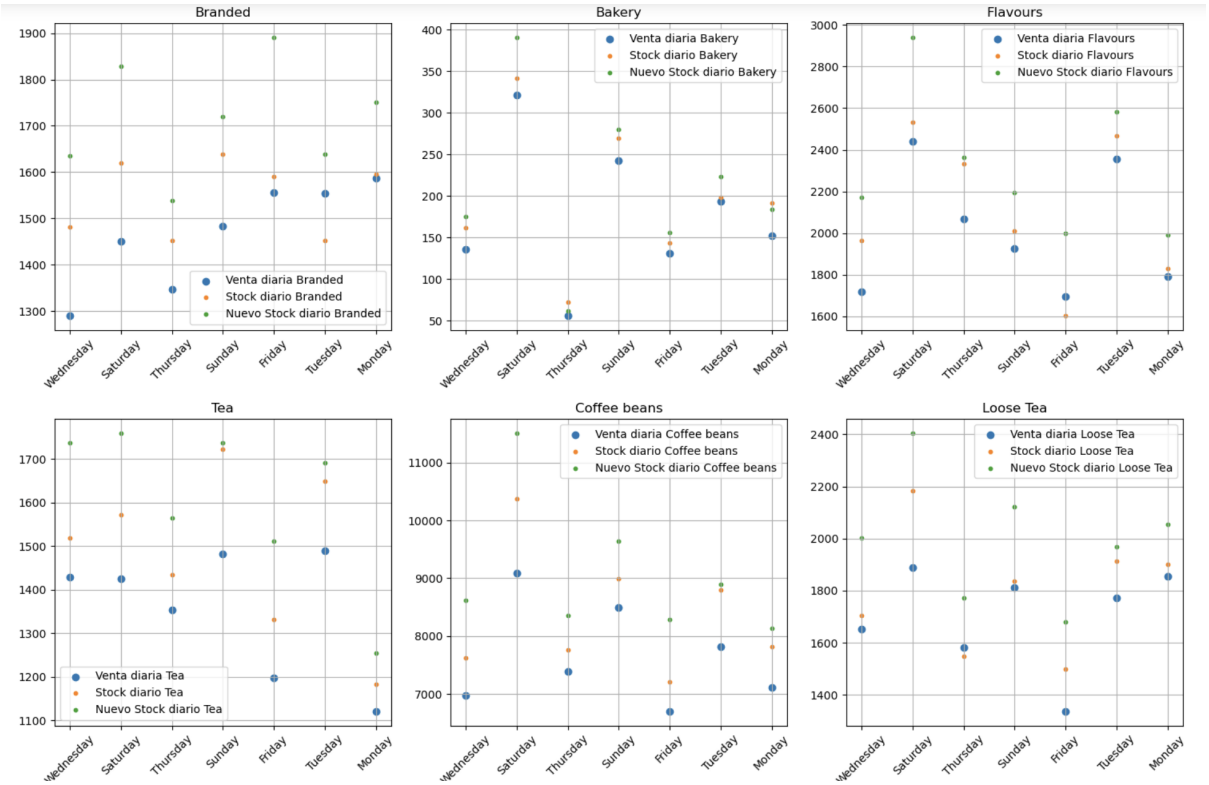
Nuestro objetivo será responder a cuatro preguntas que nos podrán orientar sobre cómo enfocar la venta de los productos de nuestra cafetería, dependiendo de la hora y del día de la semana. Comenzamos realizando una descripción exhaustiva del dataset. En ella podemos ver características como el día de la semana y el tipo de producto que más se repite, por ejemplo o, también, variables estadísticas como la media o la desviación típica. En general, información suficiente para saber cómo abordar el estudio del conjunto de datos.

	transaction_date	transaction_time	customer_id	product_id	quantity	unit_price	gender	birth_year	waste	quantity_sold	c
count	28326	28326.000000	28326.000000	28326.000000	28326.000000	28326.000000	28326	28326.000000	28326.000000	28326.000000	
unique	7	nan	nan	nan	nan	nan	2	nan	nan	nan	
top	Monday	nan	nan	nan	nan	nan	M	nan	nan	nan	
freq	4701	nan	nan	nan	nan	nan	14264	nan	nan	nan	
mean	nan	11.442773	17.338029	48.003530	1.436454	3.439395	nan	49.005366	10.587164	17.552425	
std	nan	3.796336	56.578730	18.147488	0.550347	2.938580	nan	18.236066	4.267020	9.372819	
min	nan	1.000000	0.000000	1.000000	1.000000	0.800000	nan	18.000000	0.000000	0.000000	
25%	nan	8.000000	0.000000	33.000000	1.000000	2.500000	nan	33.000000	10.000000	9.000000	
50%	nan	10.000000	0.000000	47.000000	1.000000	3.000000	nan	49.000000	10.000000	18.000000	
75%	nan	15.000000	0.000000	60.000000	2.000000	3.750000	nan	65.000000	10.000000	26.000000	
max	nan	20.000000	306.000000	87.000000	8.000000	45.000000	nan	80.000000	47.000000	35.000000	

Para responder a la primera pregunta, ***¿Cómo optimizar el beneficio de las ventas de una cafetería?,*** lo primero que haremos será ver la relación que hay entre la venta diaria de los productos y el stock diario disponible.



Nuestro objetivo será establecer un stock fijo de manera a que si, por ejemplo, un producto vende 10 unidades diarias, tengamos a disposición 15 unidades en stock. Esto funcionará en ambos sentidos, es decir, si tenemos 30 unidades en stock, las reduciremos y si tenemos 10, las aumentaremos. Partimos del principio que las ventas son una variable que fluctúa y por ello acabaremos obteniendo un mayor beneficio, a largo plazo, si adoptamos esta estrategia.



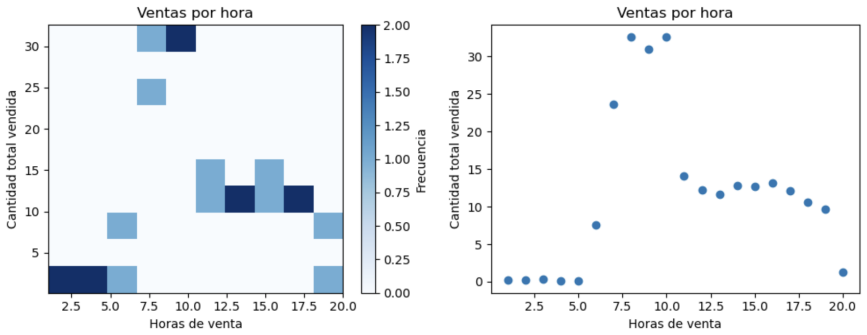
El algoritmo preestablecido para llevar a cabo este ajuste se basa en el error absoluto de las medias de la cantidad vendida y de la cantidad de stock disponible. En el caso en que la diferencia sea menor a 5, aumentaremos el stock en los productos que presenten esta condición y reduciremos en aquellos en que no lo hagan. En caso en que la diferencia sea superior a 5, el proceso será inverso, se reducirá el stock para aquellos productos que lo superen y se aumentará para aquellos que no lo hagan.

```
#Ahora vamos a aplicar una mejora en la cantidad del stock

dias = sorted(set(dataset_1['transaction_date'].values))

#stock final = venta mensual media + stock actua
dataset_1["Nuevo_stock_total"] = np.zeros(len(dataset_1))


for k in range(len(dias)):

    ventas = dataset_1[dataset_1['transaction_date'] == dias[k]]
    ventas = ventas[["transaction_date", "quantity_sold", "Stock"]]
    ventas = ventas.dropna(axis=0)
    dif = abs(ventas["Stock"].values.mean() - ventas["quantity_sold"].values.mean())
    print("")
    print(f"La diferencia entre la media del stock y de las ventas el día {dias[k]} es de {dif}")
    print("Escriba la diferencia de stock óptimo")
    x = input()
    x = float(x)

    if dif < 2:
        while  x >= abs(ventas["Stock"].values.mean() - ventas["quantity_sold"].values.mean()):

            for i in range(len(ventas)):
                if ventas["Stock"].values[i] - ventas["quantity_sold"].values[i] < (x+0.5):
                    ventas["Stock"].values[i] = ventas["Stock"].values[i] + 1

                else:
                    ventas["Stock"].values[i] = ventas["Stock"].values[i] - 1

            condicion = dataset_1["transaction_date"] == dias[k]
            dataset_1.loc[condicion, "Nuevo_stock_total"] = ventas["Stock"]
```

```
    else:

        while  abs(ventas["Stock"].values.mean() - ventas["quantity_sold"].values.mean()) >= x:

            for i in range(len(ventas)):
                if ventas["Stock"].values[i] - ventas["quantity_sold"].values[i] > (x+0.5):
                    ventas["Stock"].values[i] = ventas["Stock"].values[i] - 1

                else:
                    ventas["Stock"].values[i] = ventas["Stock"].values[i] + 1

            condicion = dataset_1["transaction_date"] == dias[k]
            dataset_1.loc[condicion, "Nuevo_stock_total"] = ventas["Stock"]

print("")
print("-------------------------------")
print("La optimización está lista")
```

La siguiente pregunta que vamos a tratar de responder va a ser doble, ***¿Cuáles son las horas en las que más se vende? y según el día y la hora, ¿Habrá alguna venta?***

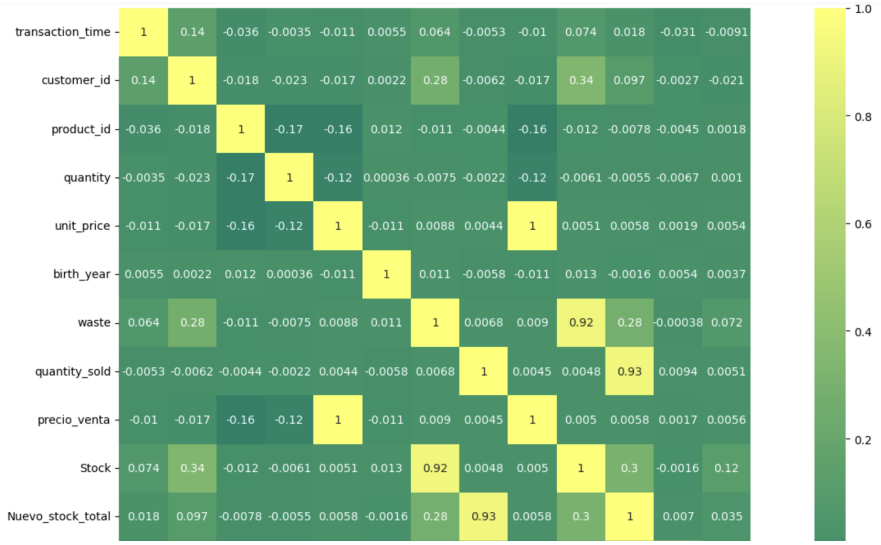
Lo primero que haremos será filtrar nuestro dataset de manera a tomar la suma de las ventas, para todos los días de la semana, según la hora. Visualizamos el resultado con dos “plots”, uno “hist2d” y otro “scatter”.



Añadir que las ventas, el eje y, han sido reducidas de escala para poder ver un gráfico más elegante.
Observamos una clara tendencia a vender productos entre las 6 y las 10 de la mañana, en el desayuno. Si bien destacamos una tendencia descendente, podemos apreciar un pico secundario a media tarde, quizás correspondiente a la típica hora del café. Este resultado vendría a indicarnos varias posibles mejoras a tener en cuenta. Una primera, aumentar el stock debido a la alta demanda, una segunda invertir en marketing o publicidad en este rango de horas, y por último, realizar un trabajo de preparación previo en las horas anteriores para poder suplir la alta demanda.

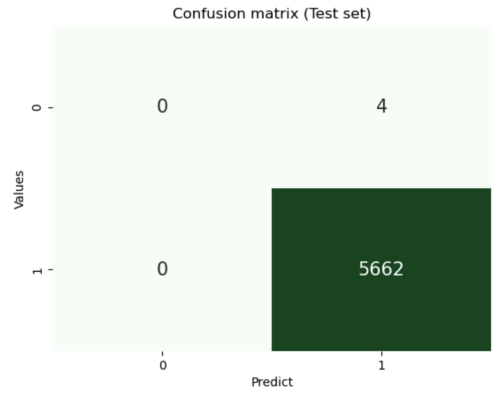
A continuación, **responderemos a la pregunta de si habrá una venta según la hora y el día que se compre**. Presentamos variables categóricas con una numérica, la hora. Usaremos el modelo "KNeighborsClassifier". Haremos uso, también, de funciones como "GridSearchCV" para mejorar nuestro modelo.

Lo primero será crear una función predicción en la que no solo incluimos el modelo y funciones de optimización pertinentes, sino que añadiremos librerías que permitirán convertir las variables categóricas en numéricas. Hablamos de "LabelEncoder". Además usaremos la matriz de correlación y de confusión para evaluar mejor nuestro modelo.




```
El modelo de random forest tiene un rendimiento de -0.0046333543357786056
El modelo de KNeighbors tiene un rendimiento de 0.9992940345923049
```

He querido incluir, en la función, el modelo Random Forest Regressor, para mostrar la diferencia entre tratar un problema con un modelo que pretende predecir una variable categórica con uno que se centra en variables numéricas.



Podemos ver una interpretación del modelo casi perfecta. No tenemos negativos reales ni falsos negativos. Si bien tenemos 4 falsos positivos, todo lo demás se ha predicho de forma adecuada.

Sin duda, este resultado no refleja la realidad. Si volvemos a la tabla descriptiva del principio, veremos una cantidad de "y" ("yes", se ha vendido un producto), demasiado grande comparada a "n". O sea, no

necesitamos de un modelo para predecir si habrá una venta o no,
ya que casi siempre la hay. Aun así este modelo sería efectivo si tuviéramos muchos más datos al respecto.

Por último, responderemos a la pregunta ***¿Qué producto es más probable que se venda?,*** según la edad, el género, la hora y el día. La estructura de la función predicción es exactamente la misma que la anterior. Cambiamos el "target", las "features" y hacemos trabajar al modelo. El resultado es:

```
--------------------

La precisión del modelo en el grupo test es de 0.5003529827038475

--------------------
El producto que consumirá una persona de género F, edad 25, el Monday a las 7 es ['Drinking Chocolate']
```

Tenemos un rendimiento del modelo bastante bajo como para apoyar nuestra estrategia de mercado en él. La pregunta resulta demasiado ambiciosa para un conjunto de datos demasiado escaso.