Project III: Coffee shop analysis

1. Objective

The aim of this project is answering 4 queries: How ***to optimize the benefits of sales of a coffee shop?***, ***What are the peak sales hours by day and time? Will there be any sales? Which product is most likely to sell based on age, gender, day, and time?***

2. Data collection

This project was charged by a specify company across feverr

3. Data preparation

i. Data cleaning

We have a dataset showing sales of a coffee shop. Initially, we filter the data, removing any null values, finally selecting the appropriate columns.

ii.LabelEncoder

We will utilise “LabelEncoder” to encode the categorical values.

4. Data exploration

i. Simple graphs

At this stage, we have already filtered, managed data, and we can start to visualize with simple graphs such a “hist2d” o “scatter”, both are in Matplotlib libraries.

ii. Non-graphical exploration

Ideally, we could utilize the pre-defined function “describe” to conduct a non-graphical analysis. Furthermore, we could personalize this depiction using “style.background_gradient” to gain a comprehensive understanding of the entire dataset.

5. Data Modeling

i. Models and variables

Once we have the data clean and ready to be submitted to the data modeling process. We choose our features and target. It's important to note that we take a supervised approach
to this problem, ensuring that the model has all the necessary information.

ii. Execution

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.

Due to the nature of our problem, we will use a classification model. We will select a range of classification models, such as KMeans, KNeighborsClassifier, and RandomForestClassifier. Finally, we will get the model which gives us the highest score.

iii. Model diagnosis and comparison

After the execution and concluding with a selection of the best model, we could use “cross_val_score” to determine the most effective way to partition our "train" and "test" sets. Additionally, we could employ “GridSearchCV” to get the hyperparameters which fit better with our data.

Furthermore,it would be appropriate to use the confusion matrix to observe, more properly, the performance of the model selected.

5. Presentation

i. Presentation

The model has already been selected, enhanced, and executed. We can present the outcome of our model visually or in tabular format.

Abstract

Our project is about a coffee shop, it consists of different datasets, getting information of sales, gender, age of customers, 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

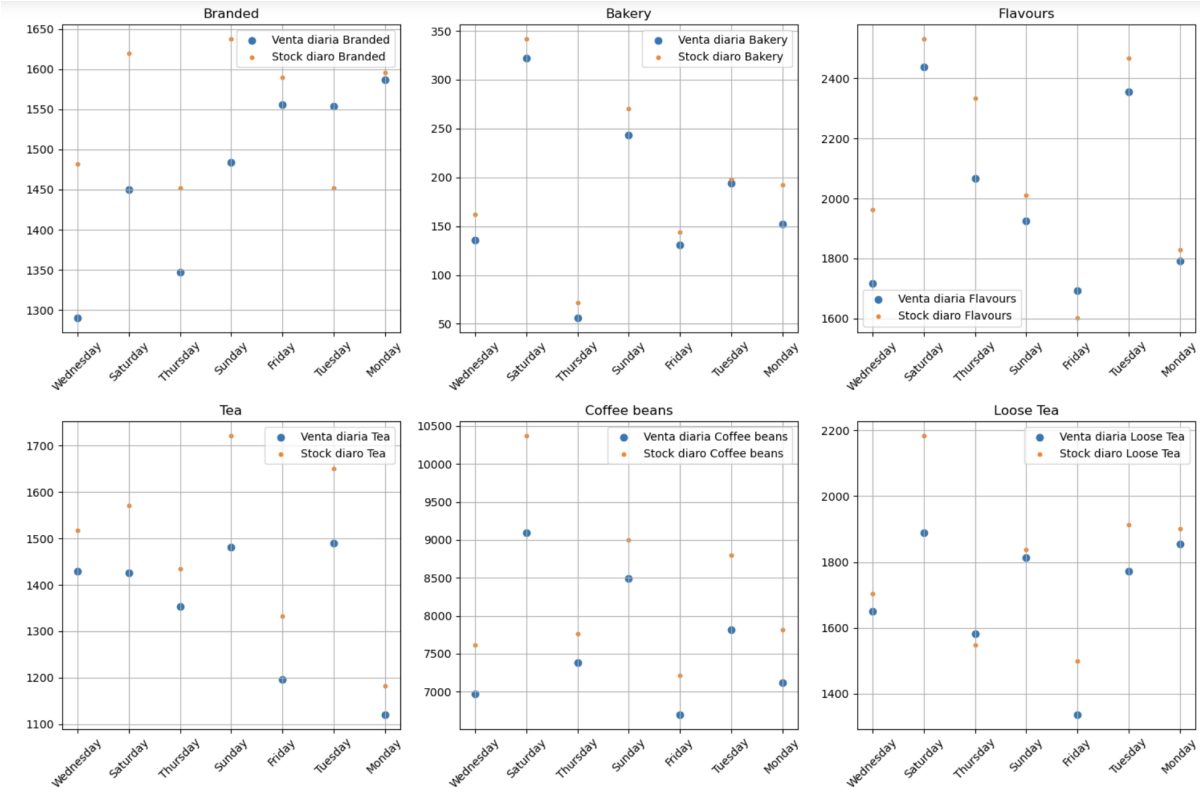
product_id	waste	quantity_sold
69	10	8
70	6	12
71	10	8
72	39	9
73	9	9

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

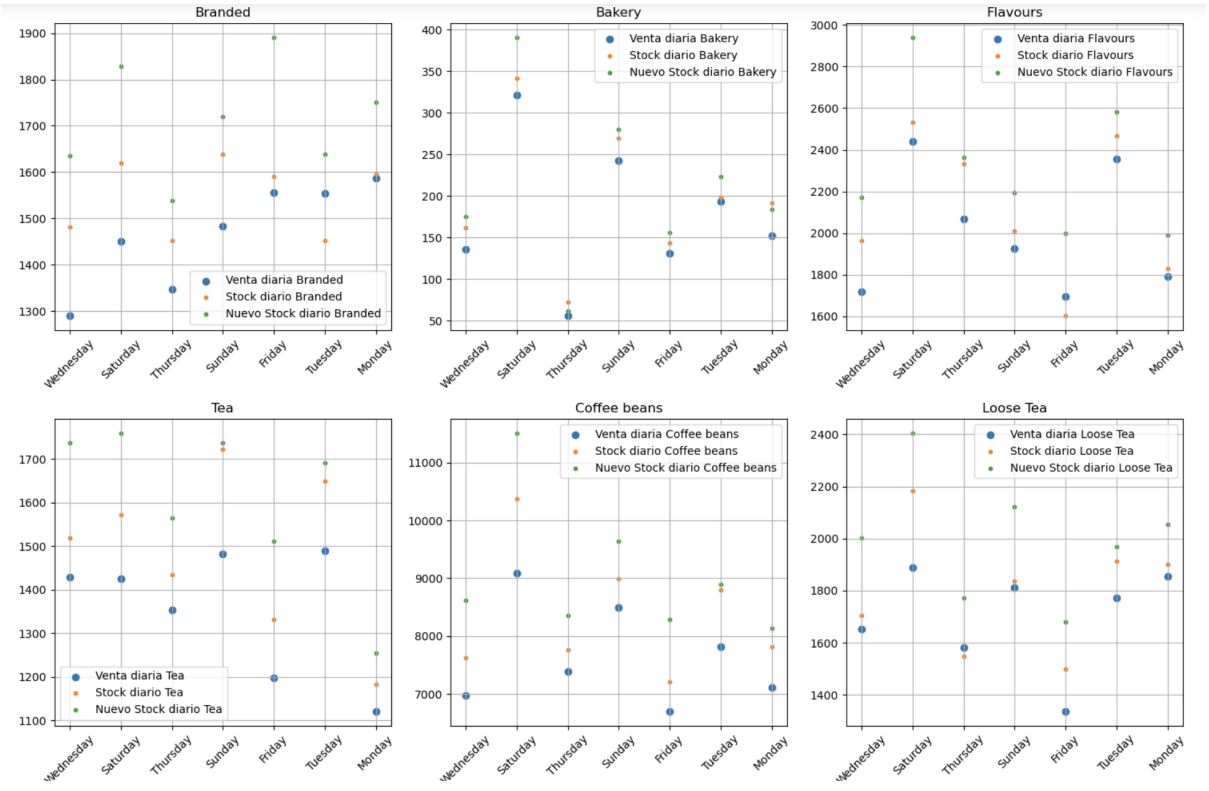
The main objective will be answering 4 queries giving some idea on how to focus the sales of our coffee shop. We start doing an exhaustive description of the dataset from a tabular format. In the table, under, we can see some statistics variables such as mean or standard deviation, as well as the product with the highest frequency.

	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	

Firstly, we can answer the next question: ***How to optimize the benefits of sales of our coffee shop?*** Initially, we shall examine the correlation between daily sales and daily inventory.



Our objective will be establishing a fixed inventory, that is, one product is purchased 10 times per day but we have 20 units of this product available to the inventory. We aim to optimize this number by reducing it. Since sales fluctuate, we expect to see greater benefits in the foreseeable future.

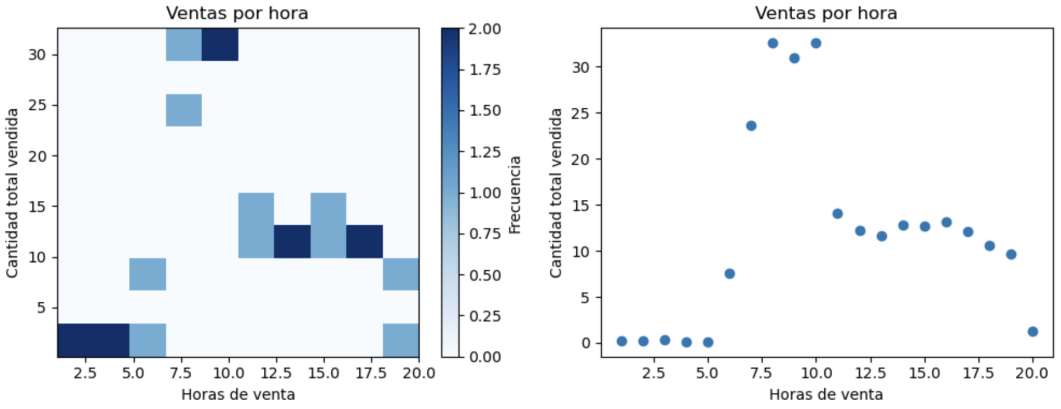


The algorithm used to fit the inventory to the sales units is based on the absolute error of this disparity. In the case of the difference being lowering 5, we will increase the amount of inventory units and we will diminish it if the converse is true.

```
#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")
```

The next stage is to answer two interrelated inquiries. ¿Which is the most sales hours?, and based on the day and the hour, Will there be any sales? We will start to manage our dataset by aggregating the sales of various products for every day of the week, sorted by hours. The visualization will be depicting by two plots, “scatter” and “hist2d”

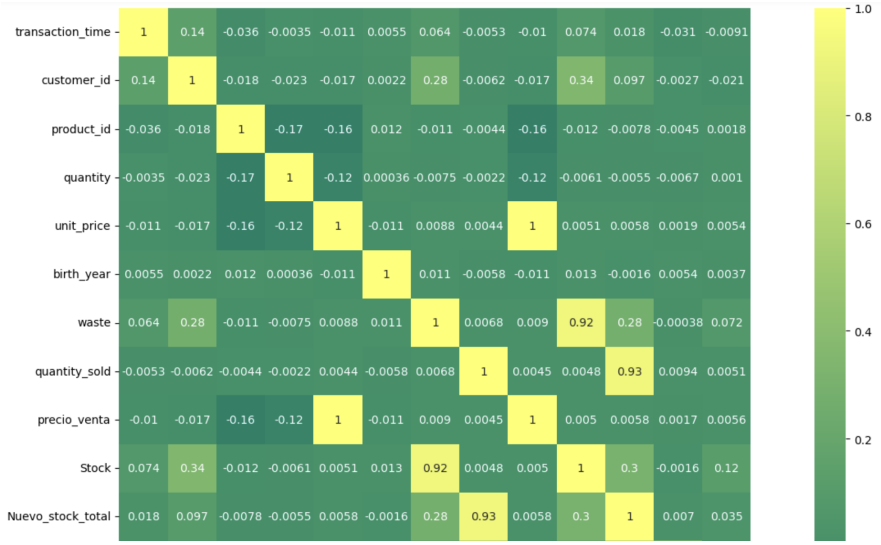


Note that the y-axis was reduced to create a more elegant graph. It is evident that there is a distinct trend in purchasing products during breakfast hours, from 6 to 10 am.Despite the graph displaying a decrease in trends, a secondary peak is observed in the afternoon, which could be attributed to the typical coffee hour.

Based on the upshot of this graph we can adopt some different strategies to enhance our business. Firstly, we can increase the inventory units because of the high demand. Secondly, invest in marketing or publicity in the peak range of hours. Finally, to carry out preparatory work in the hours beforehand to be able to meet the high demand.

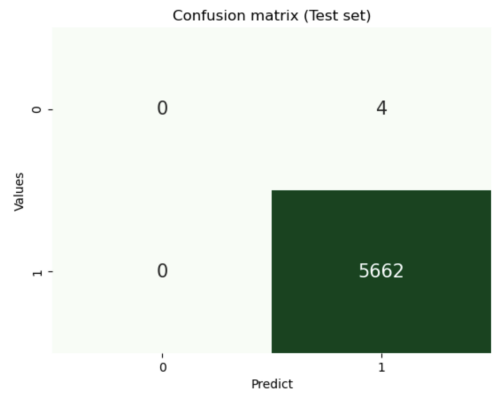
The ultimate step is answering the inquiry: ***Will there be any sales?***. We have a mix of categorical and numerical variables. We will employ the "KNeighborsClassifier" and may utilize "GridSearchCV" to enhance our model.

We create a prediction function where we introduce some functions to optimize the model and commands to convert the categorical variables to numerical ones. Finally, we have evaluated the model's performance by representing the correlation and confusion matrices.




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

Motivated to show the difference between aboard a categorical problem with a numerical model, I scripted the show of scores of random forest models with kneighbors models.



We can see an almost perfect interpretation of the model. We have no real negatives and no false negatives.Despite having four false positives, everything else has been predicted appropriately.

This result certainly does not reflect reality. If we go back to the descriptive table at the beginning, we see an amount of "y" ("yes", a product has been sold), which is too large compared to "n". In other words, we do not need a model to predict whether there will be a sale or not, since there almost always is. Even so, this model

would be effective if we had much more data on the subject.

Lastly, we aim to address the query of "***Which product has the highest likelihood of being sold, based on age, gender, time and day?***" The prediction function's structure remains identical to that aforementioned. We have refined our target audience and product features to ensure the model is fully operational. Our efforts have resulted in a successful outcome.

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

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']
```

We have a low enough model performance to support our market strategy. The question is too ambitious for too small a data set.