12 DE NOVIEMBRE DE 2023

DATA SCIENCE
COMISION 52290

SEBASTIAN GUIDO DEL OLMO

ENTREGA FINAL

INTRO

El dataset elegido es El Costo de Vida por País.

La idea de analizar este dataset es, mediante el uso de ML, tratar de encontrar que países tiene características similares para agruparlas, y posteriormente determinar qué países (dentro de los grupos) son las mejores para desarrollar y vivir, enfocado en individuos que desean buscar otros horizontes y por otro lado, con una mirada comercial, una inversión inmobiliaria según la relación entre el ingreso y el gasto de sus ciudadanos.

Con este objetivo presente, las preguntas son que grupo de países son parecidos según las variables provistas por el dataset. Posteriormente, qué ciudades tiene mejores promedios en la relación entre salario y costo de vida, para tomar decisiones según lo planteado tanto comercial como individualmente.

Generé un filtrado de algunas Países y sus ciudades repartidos en los 5 continentes para comparar y graficar mejor los datos.

Estos datos, nos van marcado patrones de precios de costo de vida muy determinantes según la región del planeta que nos encontramos y sus características socioeconómicas. Tenemos idea normalmente de estos datos pero verlos con números o gráficos, acentúan mucho más el entendimiento.

En base a esto, ya podemos decir que los países desarrollados, parecerían tener costos de vida más altos, pero la relación con el ingreso es muy superior que en los no desarrollados, por eso ya tendríamos una respuesta en cuanto a la elección de países para vivir. Está claro que sólo estamos analizando variables numéricas y no sociales que también son determinantes para este tipo de elección.

También podemos empezar a responder con respecto a un desarrollo inmobiliario (sacando datos que no conocemos como: normativas, densidad poblacional, crédito,

cantidad de ciudadanos, etc.,) que las personas que viven en un país desarrollado pueden acceder más rápidamente a pagar un crédito y liquidarlo en menor tiempo que en el otro grupo. Como en Australia con ingresos medios de $ 3500, aunque su costo por M2 sea más elevado también.

ANÁLISIS EXPLORATORIO DE DATOS (EDA)

Definición de objetivo: Qué ciudades o países son mejores para vivir según sus costos de vida y en dónde se recomienda invertir en proyecto inmobiliario?

Contexto comercial: A través de los datos sobre el costo de vida, queremos proyectar un informe de mejores países para vivir y responder a una empresa de capitales de inversión, donde poner dinero para un desarrollo de viviendas.

Problema Comercial: Existen grupos de ciudades que se diferencian de otras según sus costos de vida? Cuál es la relación entre ingreso y costo del m2 promedio en esos grupos? Podemos hacer recomendaciones de inversión sin contar con datos demográficos o información de leyes o normas del lugar? Dependiendo del riesgo del inversor, es más rentable un país en desarrollo o uno que ya lo está, analizando salario y gastos?

Contexto analítico: Extraemos los datos del dataset de costo de vida, en donde encontramos 54 variables para analizar. Debemos transformar columnas, normalizar números y nulos. Verificar datos buenos y malos. Posteriormente construímos gráficos para visualizar resultados y comparar.

Exploración de datos (EDA): como indicamos previamente, los países desarrollados tienen ingresos muy superiores a los que no lo están, aunque también observamos que el costo por m2 también es más elevado. Basándonos en esta relación, en países como Australia (usd3500) o Alemania(usd2500), un individuo podrá pagar más rápidamente un crédito que en Egipto (usd200) O Pakistán (usd200). El último gráfico de barras, donde tenemos de un lado el ingreso y el otro el costo del m2, se ve bastante armónico o relacionado, salvo en casos como China o Argentina donde el costo por m2 es mucho

más elevado visualmente que el salario. En el caso específico de Argentina, se debe a una tremenda inflación y devaluación del salario (usd300), aunque también genera una oportunidad a largo plazo de recuperación, de invertir en proyectos inmobiliarios actuales (usd2500 m2), para apostar a una recuperación futura.

VARIABLES DE BASE DE DATOS

4874 rows × 59 columns
Entender y cuantificar los costos de vida según ciudad, analizando las distintas variables ofrecidas.

- Column Description
- city Name of the city
- country Name of the country
- x1 Meal, Inexpensive Restaurant (USD)
- x2 Meal for 2 People, Mid-range Restaurant, Three-course (USD)
- x3 McMeal at McDonalds (or Equivalent Combo Meal) (USD)
- x4 Domestic Beer (0.5 liter draught, in restaurants) (USD)
- x5 Imported Beer (0.33 liter bottle, in restaurants) (USD)
- x6 Cappuccino (regular, in restaurants) (USD)
- x7 Coke/Pepsi (0.33 liter bottle, in restaurants) (USD)
- x8 Water (0.33 liter bottle, in restaurants) (USD)
- x9 Milk (regular), (1 liter) (USD)
- x10 Loaf of Fresh White Bread (500g) (USD)
- x11 Rice (white), (1kg) (USD)
- x12 Eggs (regular) (12) (USD)
- x13 Local Cheese (1kg) (USD)
- x14 Chicken Fillets (1kg) (USD)
- x15 Beef Round (1kg) (or Equivalent Back Leg Red Meat) (USD)
- x16 Apples (1kg) (USD)
- x17 Banana (1kg) (USD)
- x18 Oranges (1kg) (USD)
- x19 Tomato (1kg) (USD)
- x20 Potato (1kg) (USD)
- x21 Onion (1kg) (USD)
- x22 Lettuce (1 head) (USD)

- x23 Water (1.5 liter bottle, at the market) (USD)
- x24 Bottle of Wine (Mid-Range, at the market) (USD)
- x25 Domestic Beer (0.5 liter bottle, at the market) (USD)
- x26 Imported Beer (0.33 liter bottle, at the market) (USD)
- x27 Cigarettes 20 Pack (Marlboro) (USD)
- x28 One-way Ticket (Local Transport) (USD)
- x29 Monthly Pass (Regular Price) (USD)
- x30 Taxi Start (Normal Tariff) (USD)
- x31 Taxi 1km (Normal Tariff) (USD)
- x32 Taxi 1hour Waiting (Normal Tariff) (USD)
- x33 Gasoline (1 liter) (USD)
- x34 Volkswagen Golf 1.4 90 KW Trendline (Or Equivalent New Car) (USD)
- x35 Toyota Corolla Sedan 1.6l 97kW Comfort (Or Equivalent New Car) (USD)
- x36 Basic (Electricity, Heating, Cooling, Water, Garbage) for 85m2 Apartment (USD)
- x37 1 min. of Prepaid Mobile Tariff Local (No Discounts or Plans) (USD)
- x38 Internet (60 Mbps or More, Unlimited Data, Cable/ADSL) (USD)
- x39 Fitness Club, Monthly Fee for 1 Adult (USD)
- x40 Tennis Court Rent (1 Hour on Weekend) (USD)
- x41 Cinema, International Release, 1 Seat (USD)
- x42 Preschool (or Kindergarten), Full Day, Private, Monthly for 1 Child (USD)
- x43 International Primary School, Yearly for 1 Child (USD)
- x44 1 Pair of Jeans (Levis 501 Or Similar) (USD)
- x45 1 Summer Dress in a Chain Store (Zara, H&M, ...) (USD)
- x46 1 Pair of Nike Running Shoes (Mid-Range) (USD)
- x47 1 Pair of Men Leather Business Shoes (USD)
- x48 Apartment (1 bedroom) in City Centre (USD)
- x49 Apartment (1 bedroom) Outside of Centre (USD)
- x50 Apartment (3 bedrooms) in City Centre (USD)
- x51 Apartment (3 bedrooms) Outside of Centre (USD)
- x52 Price per Square Meter to Buy Apartment in City Centre (USD)
- x53 Price per Square Meter to Buy Apartment Outside of Centre (USD)
- x54 Average Monthly Net Salary (After Tax) (USD)
- x55 Mortgage Interest Rate in Percentages (%), Yearly, for 20 Years Fixed-Rate
- data_quality 0 if Numbeo considers that more contributors are needed to increase data quality, else 1

RENOMBRADO DE COLUMNAS

df_livingcost.columns = ['Index', 'City','Country', 'Meal', 'Mealx2', 'McMeal', 'DomesticBeer', 'ImportedBeer', 'Capuccino', 'Coke', 'Water', 'Milk', 'Bread', 'Rice', 'Eggs', 'Cheese', 'Chicken', 'Beef', 'Apple', 'Banana', 'Oranges', 'Tomato', 'Potato', 'Onion', 'Lettuce', 'Water', 'Wine', 'DomesticBeer', 'ImportedBeer', 'Cigarettes', 'Ticket', 'MonthlyPass', 'TaxiStart', 'Taxi1km', 'Taxi1hour', 'Gasoline', 'VolkswagenGolf', 'ToyotaCorolla', 'BasicApartment', 'Prepaid1m', 'Internet', 'FitnessClub', 'Tennis', 'Cinema', 'Preschool', 'PrimarySchool', 'Jeans', 'Dress', 'Nike', 'Shoes', 'Apart1', 'Apart1out', 'Apart3', 'Apart3out', 'BuyApartm2', 'BuyApartm2out', 'Salary', 'Rate', 'DataQuality

```
[8] df_livingcost
```

	City	Country	Meal	Mealx2	McMeal	DomesticBeer	ImportedBeer	Capuccino	Coke	Water	...	Shoes	Apart1	Apart1out	Apart3	Apart3out	BuyApartm2	BuyApartm2out	Salary	Rate	DataQuality
0	Delhi	India	4.90	22.04	4.28	1.84	3.67	1.78	0.48	0.19	...	36.26	223.87	133.38	596.16	325.82	2619.46	1068.90	586.35	7.96	1.0
1	Shanghai	China	5.59	40.51	5.59	1.12	4.19	3.96	0.52	0.32	...	121.19	1080.07	564.30	2972.57	1532.23	17333.09	9174.88	1382.83	5.01	1.0
2	Jakarta	Indonesia	2.54	22.25	3.50	2.02	3.18	2.19	0.59	0.27	...	80.32	482.85	270.15	1117.69	584.37	2694.05	1269.44	483.19	9.15	1.0
3	Manila	Philippines	3.54	27.40	3.54	1.24	1.90	2.91	0.93	0.51	...	61.82	559.52	281.78	1754.40	684.81	3536.04	2596.44	419.02	7.80	1.0
4	Seoul	South Korea	7.16	52.77	6.03	3.02	4.52	3.86	1.46	0.78	...	108.30	809.83	583.60	2621.05	1683.74	21847.94	10832.90	2672.23	3.47	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4869	Peterborough	Australia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0
4870	Georgetown	Australia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0
4871	Ixtapa Zihuatanejo	Mexico	5.19	31.13	12.97	0.99	NaN	1.82	0.62	0.42	...	103.78	415.11	259.44	518.89	415.11	NaN	NaN	NaN	NaN	0.0
4872	Iqaluit	Canada	29.78	74.61	13.77	6.70	8.93	3.72	3.54	4.10	...	NaN	NaN	NaN	2978.11	2978.11	NaN	NaN	NaN	6.53	0.0
4873	Neiafu	Tonga	NaN	29.53	10.55	10.55	NaN	NaN	2.11	2.11	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0

4874 rows × 58 columns

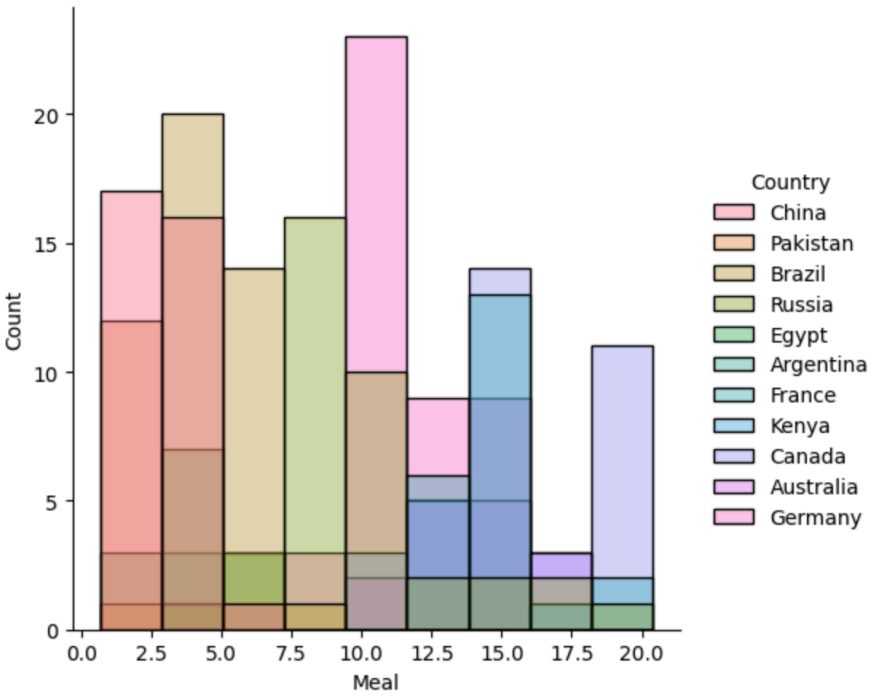
SELECCIÓN PAÍSES PARA ANALIZAR

Realizo una selección de países para hacer análisisdf_seleccion = df_livingcost.loc[df_livingcost['Country'].isin(('Argentina', 'Australia', 'Canada', 'Brazil', 'France', 'Germany', 'China', 'Pakistan', 'Egypt', 'Kenya', 'Russia'))]# df_seleccion = df_livingcost.loc[df_livingcost['Country'] == 'Argentina']

GRAFICOS

```
sns.displot(data=df_seleccion, x="Meal", hue="Country", alpha = 0.4 )
```

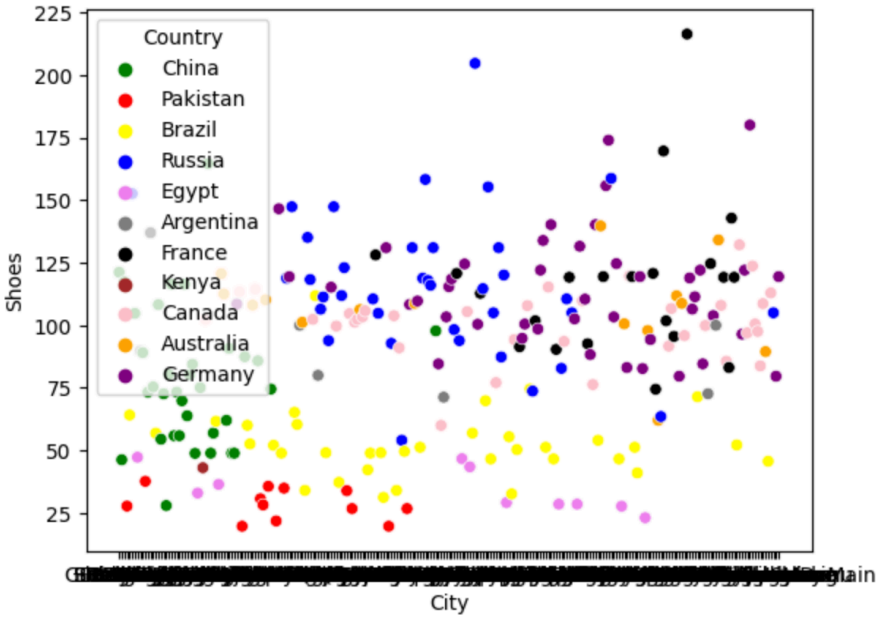
```
<seaborn.axisgrid.FacetGrid at 0x7a79b8680130>
```



En el histograma, los precios de la comidas, se vuelven hacia la izquierda, con valores hasta los 15 dólares y algunos pocos extendidos, justamente los países desarrollados, donde los costos de vida son altos, como Australia o Alemania. En los países menos desarrollados, generalmente el costo del alimento es bajo, pero también son los salarios. La relación entre ellos se observa por ejemplo en Alemania con un costo promedio de una comida en usd 18 los salarios son de usd 2600, en cambio en Egipto está en usd 10 pero los salarios rondan los bajísimos usd 200. Esto se replica en ambos grupos (desarrollados y no desarrollados).

```
[ ]  sns.scatterplot(data=df_seleccion, x="City", y="Shoes", hue = 'Country', palet
```

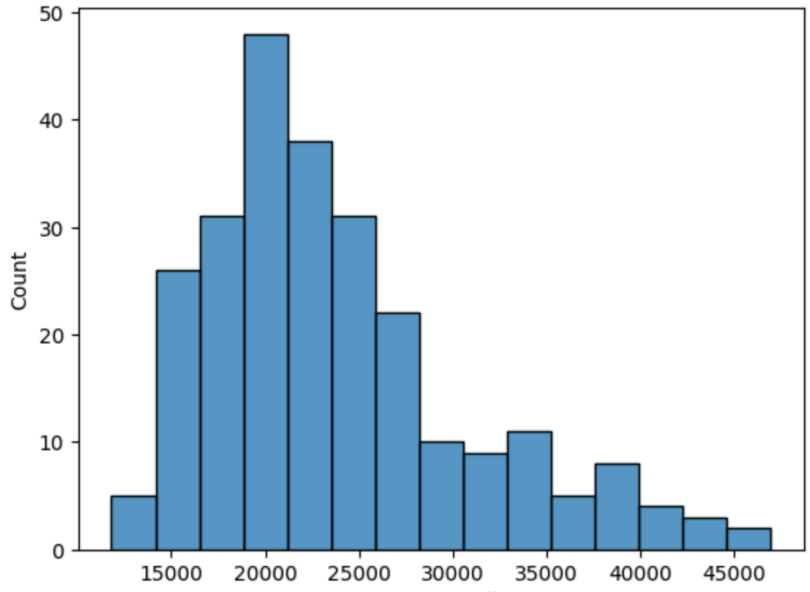
```
<Axes: xlabel='City', ylabel='Shoes'>
```



Se observa en cada uno los movimientos de los precios en dólares según las regiones, por ejemplo en el de dispersión usé la variable Zapatos, donde Pakistán tiene el menor precio (30 usd.) entiendo que por ser un país muy enfocado en la fabricación textil con mano de obra muy barata.

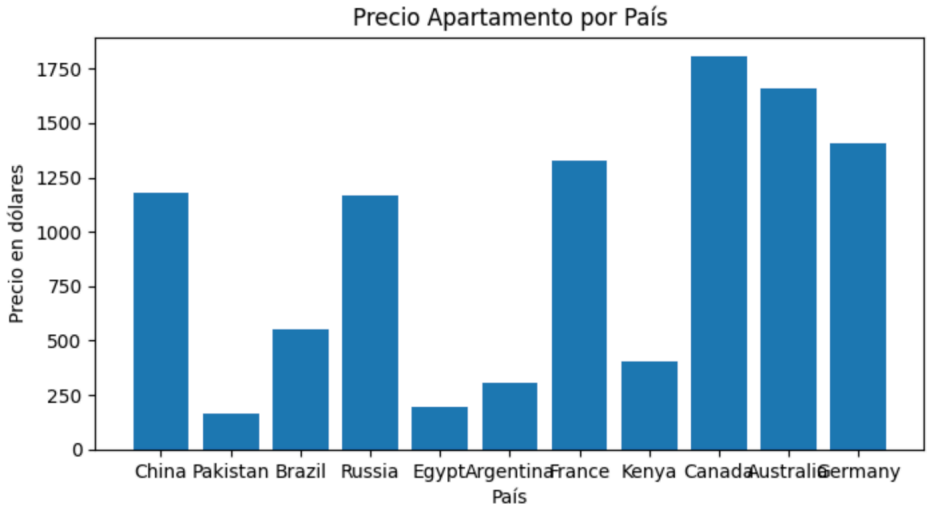
```
[ ] sns.histplot(data=df_seleccion, x="ToyotaCorolla")
```

```
<Axes: xlabel='ToyotaCorolla', ylabel='Count'>
```




```
fig, ax = plt.subplots(figsize=(8,4))
ax.bar(df_seleccion.Country, df_seleccion.Apart1)
ax.set_title('Precio Apartamento por País')
ax.set_ylabel('Precio en dólares')
ax.set_xlabel('País')
```

```
Text(0.5, 0, 'País')
```

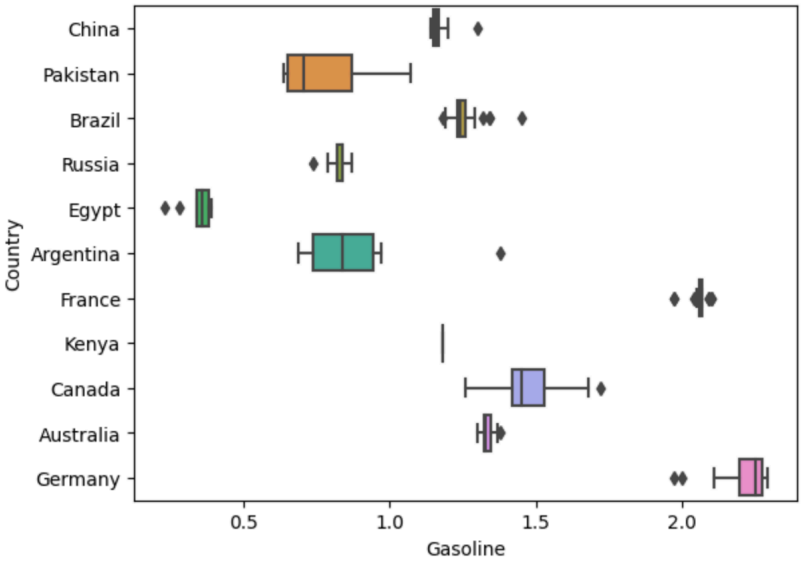


Por último, un gráfico de barras, donde medimos el precio del alquiler de un departamento en el centro, y nos arroja similares datos, con gran división entre países

desarrollados con precio elevados como Australia cerca de los usd 3500 y en el otro extremo con Pakistán o Egipto, con valores entre usd 100 y usd 200.

```
[ ]  sns.boxplot(x="Gasoline", y="Country", data=df_seleccion)
```

```
<Axes: xlabel='Gasoline', ylabel='Country'>
```



Otro caso a destacar, en Argentina (subsidio del gobierno) o Egipto con un precio de la gasolina muy bajo, Alemania y Francia muy alto, por encima de los usd 2.

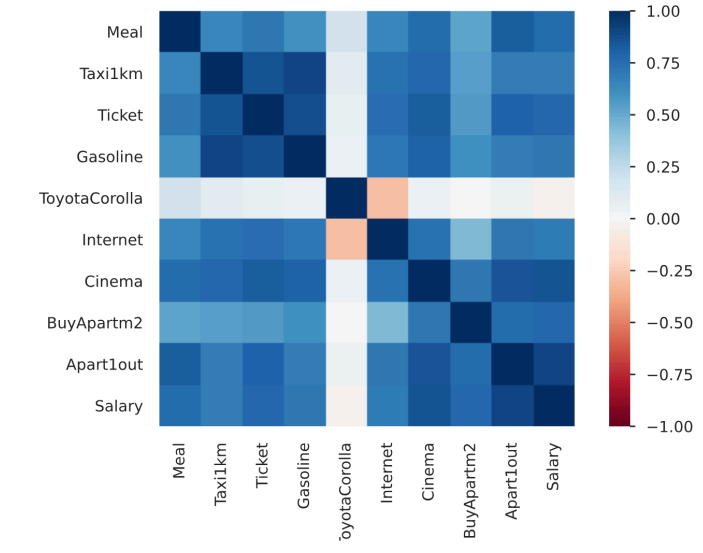
Los países desarrollados tienen ingresos muy superiores a los que no lo están, aunque también observamos que el costo por m2 también es más elevado. Basándonos en esta relación, en países como Australia (usd3500) o Alemania(usd2500), un individuo podrá pagar más rápidamente un crédito que en Egipto (usd200) O Pakistán (usd200). El último gráfico de barras, donde tenemos de un lado el ingreso y el otro el costo del m2, se ve bastante armónico o relacionado, salvo en casos como China o Argentina donde el costo por m2 es mucho más elevado visualmente que el salario. En el caso específico de Argentina, se debe a una tremenda inflación y devaluación del salario (usd300), aunque también genera una oportunidad a largo plazo de recuperación, de invertir en proyectos inmobiliarios actuales (usd2500 m2), para apostar a una recuperación futura.

DATA PROFILING (ANALISIS EXTRA)

Selecciono algunas variables para analizar con Pandas Profiling, debido al tamaño enorme que se genera al meter todas las variables.

Se puede observar tanto en Pandas Profiling, con algunas variables y en el Heatmap con todas sus variables, una buena correlación entre las variables que fueron previamente seleccionadas a través de los algoritmos de selección.

Estas variable son Salary, BuyApartm2 y una extra que no utilicé, que es Meal.



LIMPIEZA DE DATOS

- Se realizó previamente un renombrado y eliminación de columna
- ELIMINACION DE NULOS
- REINDEXACION
- ELIMINA COLUMNAS Index y DataQuality
- SUMA DE COLUMNAS (gastos similares y reduzco cantidad de variables)

PREPARACION GRUPOS ENTRENAMIENTO

```
[ ]  # división de grupos
     ss = ShuffleSplit(n_splits=1, random_state=0, test_size=0.3)
     for train_index, test_index in ss.split(df_reindexado2):
         print(len(train_index))
         print(len(test_index))
     df_reindexado2['Train'] = 0
     df_reindexado2.loc[df_reindexado2.index.isin(train_index),'Train'] = 1
```

```
884
380
```

ELIMINACION DE ATIPICOS

```
[ ]  # Eliminar atípicos generales por LocalOutlierFactor
     outlierDetector = LocalOutlierFactor(n_neighbors = 2, metric = "manhattan", contamination = 0.02, novelty = True)
     outlierDetector.fit(df_reindexado2.loc[df_reindexado2['Train'] == 1, numericas])
     ind_train = outlierDetector.predict(df_reindexado2[numericas])
     df_reindexado3 = df_reindexado2.loc[ind_train == 1]
```