

Informe de resultados caso Telcom X

Introducción

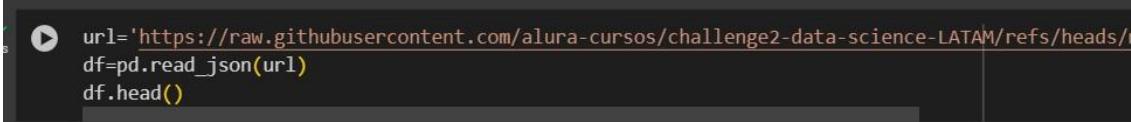
Telcom X es una empresa de telecomunicaciones, el cual a solicitado mediante un proyecto el análisis de la permanencia de los clientes en sus servicios llamado “Churn de clientes”, derivado que la empresa esta enfrentando un índice alto de evasión de sus clientes y aun no han identificado el problema de esta evasión, para ellos se han proporcionado datos para realizar un análisis exploratorio para de esta manera, podría realizar un análisis predictivo y determinar de donde provienen esta evasión de clientes.

Limpieza y Tratamiento de Datos

Para iniciar el análisis de nuestra información de estos datos, utilizaremos las siguientes bibliotecas en nuestro notebook pandas, matplotlib y seaborn, y en caso de que necesitemos cálculos o graficas clasificatorias agregamos numpy y plotly.

1. Exportación

Utilizamos la propiedad de pandas para leer el archivo json mediante la variable url, para así poder llamar y visualizar los datos, llamando la los 5 primeros de la tabla donde vemos que tiene listas simples en las columnas categorías y numéricas



```
urllib.request.urlretrieve("https://raw.githubusercontent.com/alura-cursos/challenge2-data-science-LATAM/refs/heads/main/telco-churn.json", "telco-churn.json")
df=pd.read_json(url)
df.head()
```

	customerID	Churn	customer	phone	internet	account
0	0002-ORFB0	No	{'gender': 'Female', 'SeniorCitizen': 0, 'Part...	{'PhoneService': 'Yes', 'MultipleLines': 'No'}	{'InternetService': 'DSL', 'OnlineSecurity': '...}	{'Contract': 'One year', 'PaperlessBilling': '...}
1	0003-MKNFE	No	{'gender': 'Male', 'SeniorCitizen': 0, 'Partne...	{'PhoneService': 'Yes', 'MultipleLines': 'Yes'}	{'InternetService': 'DSL', 'OnlineSecurity': '...}	{'Contract': 'Month-to-month', 'PaperlessBilli...}
2	0004-TLHLJ	Yes	{'gender': 'Male', 'SeniorCitizen': 0, 'Partne...	{'PhoneService': 'Yes', 'MultipleLines': 'No'}	{'InternetService': 'Fiber optic', 'OnlineSecu...}	{'Contract': 'Month-to-month', 'PaperlessBilli...}
3	0011-IGKFF	Yes	{'gender': 'Male', 'SeniorCitizen': 1, 'Partne...	{'PhoneService': 'Yes', 'MultipleLines': 'No'}	{'InternetService': 'Fiber optic', 'OnlineSecu...}	{'Contract': 'Month-to-month', 'PaperlessBilli...}
4	0013-EXCHZ	Yes	{'gender': 'Female', 'SeniorCitizen': 1, 'Part...	{'PhoneService': 'Yes', 'MultipleLines': 'No'}	{'InternetService': 'Fiber optic', 'OnlineSecu...}	{'Contract': 'Month-to-month', 'PaperlessBilli...}

Ahora vemos los tipo de datos que tenemos mediante la función info para así saber si son objetos, número enteros y números decimales

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7267 entries, 0 to 7266
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   customerID  7267 non-null    object  
 1   Churn        7267 non-null    object  
 2   customer     7267 non-null    object  
 3   phone         7267 non-null    object  
 4   internet     7267 non-null    object  
 5   account       7267 non-null    object  
dtypes: object(6)
memory usage: 340.8+ KB
```

2.Transformacion

Lo primero que devmeos hacer es normalizar nuestros datos, desglosando las listas simples que tenemos en nuestras columnas categóricas y columnas numéricas de la siguiente manera

```
df_normalizado_customer = pd.json_normalize(
    data=df.to_dict(orient='records'),
    record_path=None,
    meta=['customerID', 'Churn'],
    errors='ignore')
```

donde le indicacmos a nuestra variable que normalizara los datos de la variable “df” que se encuentren en las columnas con listas simples, así mismo indicamso que no se encuentran columnas anidadas al especificar “None”, agregamos las dos columnas que no contienen listas simples que son solo un conjunto de datos en seria del dataframe, por ultimo le damos ignorar error para así cuando encuentre alguna clave mencionada en meta, no lance ningún error y solo la ignore. Así mismo realizamos una lista con los nombres de las columnas para renombrarlas utilizando la función reanem

J.G.C

```

nuevos_nombres = {
    'customer.gender': 'genero del cliente',
    'customer.SeniorCitizen': 'cliente senior',
    'customer.Partner': 'cliente con pareja',
    'customer.Dependents': 'cliente con dependientes',
    'phone.PhoneService': 'cliente con servicio telefonico',
    'phone.MultipleLines': 'cliente con multiples lineas',
    'internet.InternetService': 'servicio de internet',
    'internet.OnlineSecurity': 'seguridad en linea',
    'internet.OnlineBackup': 'backup en linea',
    'internet.DeviceProtection': 'proteccion de dispositivo',
    'internet.TechSupport': 'soporte tecnico',
    'internet.StreamingTV': 'streaming tv',
    'internet.StreamingMovies': 'streaming movies',
    'account.Contract': 'Tipo de contrato',
    'account.PaperlessBilling': 'Facturacion en linea',
    'account.PaymentMethod': 'Metodo de pago',
    'account.Charges.Monthly': 'costo mensual',
    'account.Charges.Total': 'costo total',
    'customer.tenure': 'Meses de contrato del cliente'
}

```

```

df_normalizado_customer=df_normalizado_customer.rename(columns=nuevos_nombres)
df_normalizado_customer.head()

```

	customerID	churn	genero del cliente	cliente senior	cliente con pareja	cliente con dependientes	Meses de contrato del cliente	cliente con servicio telefonico	cliente con multiples lineas	servicio de internet	... backup en linea	proteccion de dispositivo	soporte tecnico	streaming tv	streaming movies	Tipo de contrato	Facturacion en linea	Metodo de pago	costo mensual	
0	0002-ORFB0	No	Female	0	Yes	Yes	9	Yes	No	DSL	...	Yes	No	Yes	Yes	No	One year	Yes	Mailed check	65
1	0003-MKNFE	No	Male	0	No	No	9	Yes	Yes	DSL	...	No	No	No	No	Yes	Month-to-month	No	Mailed check	59
2	0004-TLHLJ	Yes	Male	0	No	No	4	Yes	No	Fiber optic	...	No	Yes	No	No	No	Month-to-month	Yes	Electronic check	73
3	0011-IGKFF	Yes	Male	1	Yes	No	13	Yes	No	Fiber optic	...	Yes	Yes	No	Yes	Yes	Month-to-month	Yes	Electronic check	98
4	0013-EXCHZ	Yes	Female	1	Yes	No	3	Yes	No	Fiber optic	...	No	No	Yes	Yes	No	Month-to-month	Yes	Mailed check	83

Ahora tenemos nuestra dataframe en lista con los datos desglosados y los nombres de la columna para una mayor visualización, esto lo visualizamos con las columnas llamándola en formato lista , y llamamos la información en info para saber los tipos de datos que tenemos para su posterior transformación en caso de ser requerida

J.G.C

```
columns_df=list(df_normalizado_customer.columns)
columns_df

['customerID',
 'Churn',
 'genero del cliente',
 'cliente senior',
 'cliente con pareja',
 'cliente con dependientes',
 'Meses de contrato del cliente',
 'cliente con servicio telefonico',
 'cliente con multiples lineas',
 'servicio de internet',
 'seguridad en linea',
 'backup en linea',
 'proteccion de dispositivo',
 'soporte tecnico',
 'streaming tv',
 'streaming movies',
 'Tipo de contrato',
 'Facturacion en linea',
 'Metodo de pago',
 'costo mensual',
 'costo total']
```

J.G.C

```
df_normalizado_customer.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7267 entries, 0 to 7266
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   customerID      7267 non-null    object  
 1   Churn            7267 non-null    object  
 2   genero del cliente 7267 non-null    object  
 3   cliente senior   7267 non-null    int64  
 4   cliente con pareja 7267 non-null    object  
 5   cliente con dependientes 7267 non-null    object  
 6   Meses de contrato del cliente 7267 non-null    int64  
 7   cliente con servicio telefonico 7267 non-null    object  
 8   cliente con multiples lineas 7267 non-null    object  
 9   servicio de internet 7267 non-null    object  
 10  seguridad en linea 7267 non-null    object  
 11  backup en linea 7267 non-null    object  
 12  proteccion de dispositivo 7267 non-null    object  
 13  soporte tecnico 7267 non-null    object  
 14  streaming tv     7267 non-null    object  
 15  streaming movies 7267 non-null    object  
 16  Tipo de contrato 7267 non-null    object  
 17  Facturacion en linea 7267 non-null    object  
 18  Metodo de pago   7267 non-null    object  
 19  costo mensual   7267 non-null    float64 
 20  costo total     7267 non-null    object  
dtypes: float64(1), int64(2), object(18)
memory usage: 1.2+ MB
```

Ahora al checar el tipo de datos vemos que los costos totales nos lo marca como texto y no como numero entero o decimal, por lo tanto lo transformaremos en tipo float así mismo cambiaremos el valor 0 y 1 de la columna senior para así establecer en categoría menor a 65 años y mayor a 65 años

```
df_normalizado_customer['costo total']=df_normalizado_customer['costo total'].replace(' ',0.0)
df_normalizado_customer['costo total']=df_normalizado_customer['costo total'].astype(np.float64)
df_normalizado_customer['cliente senior']=df_normalizado_customer['cliente senior'].replace(0,'menor a 65 años')
.replace(1,'mayor a 65 años')
```

Una ves realizado estos cambios volvemos a checar la información

J.G.C

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7267 entries, 0 to 7266
Data columns (total 21 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   customerID      7267 non-null   object  
 1   Churn            7267 non-null   object  
 2   genero del cliente 7267 non-null   object  
 3   cliente senior    7267 non-null   object  
 4   cliente con pareja 7267 non-null   object  
 5   cliente con dependientes 7267 non-null   object  
 6   Meses de contrato del cliente 7267 non-null   int64  
 7   cliente con servicio telefonico 7267 non-null   object  
 8   cliente con multiples lineas 7267 non-null   object  
 9   servicio de internet 7267 non-null   object  
 10  seguridad en linea 7267 non-null   object  
 11  backup en linea 7267 non-null   object  
 12  proteccion de dispositivo 7267 non-null   object  
 13  soporte tecnico 7267 non-null   object  
 14  streaming tv 7267 non-null   object  
 15  streaming movies 7267 non-null   object  
 16  Tipo de contrato 7267 non-null   object  
 17  Facturacion en linea 7267 non-null   object  
 18  Metodo de pago 7267 non-null   object  
 19  costo mensual 7267 non-null   float64 
 20  costo total 7267 non-null   float64 
dtypes: float64(2), int64(1), object(18)
memory usage: 1.2+ MB
```

Ahora vemos que nuestros costos mensuales es de tipo float, derivado que es una variable numérica de cantidad monetaria.

J.G.C

3.Buscando datos nulos o vacíos

Es hora de verificar si tenemos datos nulos o vacíos, primero desplegamos los valores con visualización de datos de las columnas para comprobar que no tenemos datos repetidos

```
for col in df_normalizado_customer.columns:  
    print(col)  
    print(df_normalizado_customer[col].unique())
```

```
customerID  
['0002-ORFBO' '0003-MKNFE' '0004-TLHLJ' ... '9992-UJOEL' '9993-LHIEB'  
'9995-HOTOH']  
Churn  
['No' 'Yes' '']  
genero del cliente  
['Female' 'Male']  
cliente senior  
['menor a 65 años' 'mayor a 65 años']  
cliente con pareja  
['Yes' 'No']  
cliente con dependientes  
['Yes' 'No']  
Meses de contrato del cliente  
[ 9 4 13 3 71 63 7 65 54 72 5 56 34 1 45 50 23 55 26 69 11 37 49 66  
67 20 43 59 12 27 2 25 29 14 35 64 39 40 6 30 70 57 58 16 32 33 10 21  
61 15 44 22 24 19 47 62 46 52 8 60 48 28 41 53 68 51 31 36 17 18 38 42  
0]  
cliente con servicio telefonico  
['Yes' 'No']  
cliente con multiples lineas  
['No' 'Yes' 'No phone service']  
servicio de internet  
['DSL' 'Fiber optic' 'No']  
seguridad en linea  
['No' 'Yes' 'No internet service']  
backup en linea  
['Yes' 'No' 'No internet service']  
proteccion de dispositivo  
['No' 'Yes' 'No internet service']  
soporte tecnico  
['Yes' 'No' 'No internet service']  
streaming tv  
['Yes' 'No' 'No internet service']  
streaming movies  
['No' 'Yes' 'No internet service']  
Tipo de contrato  
['One year' 'Month-to-month' 'Two year']  
Facturacion en linea  
['Yes' 'No']  
Metodo de pago  
['Mailed check' 'Electronic check' 'Credit card (automatic)'  
'Bank transfer (automatic)']  
costo mensual  
[65.6 59.9 73.9 ... 91.75 68.8 67.85]  
costo total  
[ 593.3 542.4 280.85 ... 742.9 4627.65 3707.6 ]
```

Done observamos que tenemos valores vacíos en nuestra columna Churn , pero necesitamos también saber cuantos tenemos y verificar como extra los nulos dado que al ver nuestra lista anterior no vemos este tipo de datos

```
"df_normalizado_customer.isnull().sum()" "(df_normalizado_customer == ' ').sum()  
"(df_normalizado_customer == "").sum()"
```

J.G.C

Al usar las anteriores líneas de código podemos obtener la sumatoria de cada una de las columnas de nuestro dataframe de valores vacíos, nulos o doble espacio

Doble espacio	Datos nulos	Datos con espacio en blanco
0	0	0
customerID	0	customerID
Churn	0	Churn
genero del cliente	0	genero del cliente
cliente senior	0	cliente senior
cliente con pareja	0	cliente con pareja
cliente con dependientes	0	cliente con dependientes
Meses de contrato del cliente	0	Meses de contrato del cliente
cliente con servicio telefonico	0	cliente con servicio telefonico
cliente con multiples lineas	0	cliente con multiples lineas
servicio de internet	0	servicio de internet
seguridad en linea	0	seguridad en linea
backup en linea	0	backup en linea
proteccion de dispositivo	0	proteccion de dispositivo
soporte tecnico	0	soporte tecnico
streaming tv	0	streaming tv
streaming movies	0	streaming movies
Tipo de contrato	0	Tipo de contrato
Facturacion en linea	0	Facturacion en linea
Metodo de pago	0	Metodo de pago
costo mensual	0	costo mensual
costo total	0	costo total

Visualizamos que tenemos 224 espacios en blanco en la columna Churn que se traduce como estado de clientes que desconocemos “si se fueron, se quedaron o simplemente los datos se dañaron por lo tanto los dejamos como “unkown”, transformándolos de la siguiente forma

```
df_normalizado_customer['Churn'].unique()
array(['No', 'Yes', ''], dtype=object)

205] df_normalizado_customer['Churn']=df_normalizado_customer['Churn'].replace('','unkown')

279] df_normalizado_customer['Churn'].unique()
array(['No', 'Yes', 'unkown'], dtype=object)
```

J.G.C

customerID	0
Churn	0
genero del cliente	0

Visualizamos que ya no tenemos datos vacíos y ahora vemos que aparece "unkown" como dato en nuestra columna

4.Transformando en minúsculas y creando una nueva columna

Como paso final de la transformación de datos pasaremos nuestros datos a minúsculas para un mejor análisis, junto que vamos a agregar la columna gastos diarios derivada de los gastos mensuales

```
for col in df_normalizado_customer.select_dtypes(include='object').columns:  
    df_normalizado_customer[col]=df_normalizado_customer[col].str.lower().str.strip()  
  
df_normalizado_customer.head()  
  
df_normalizado_customer['costos diarios']=df_normalizado_customer['costo mensual']/30
```

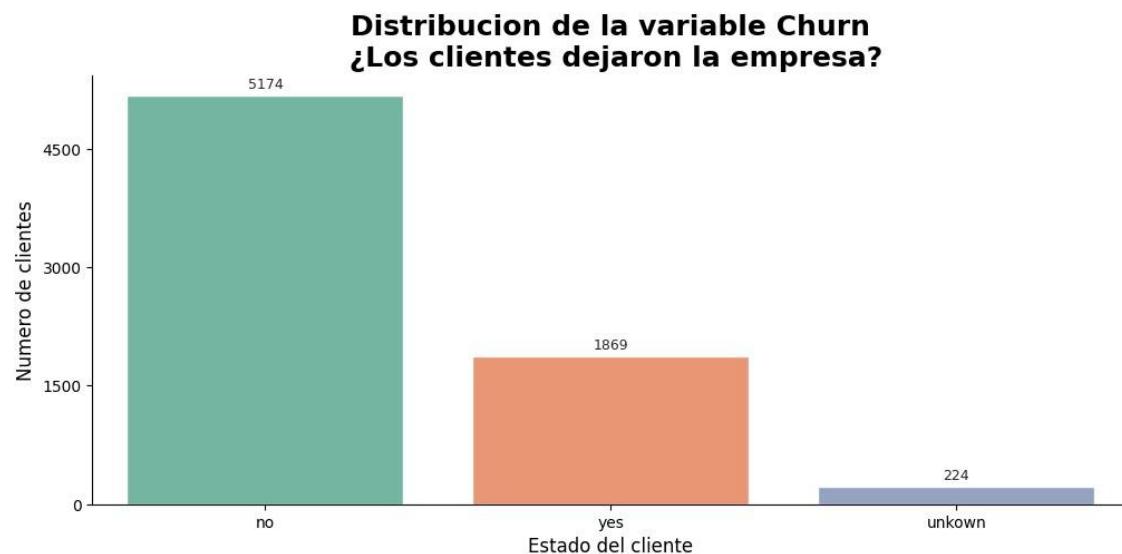
Metodo de pago	costo mensual	costo total	costos diarios
mailed check	65.6	593.30	2.186667
mailed check	59.9	542.40	1.996667
electronic check	73.9	280.85	2.463333
electronic check	98.0	1237.85	3.266667
mailed check	83.9	267.40	2.796667

Carga y análisis de los datos

A continuación solo visualizaremos las gráficas y los respectivos análisis de estas para saber como nuestros datos se desplegaban de acuerdo a las evasiones de los clientes con las diferentes categorías que tenemos

J.G.C

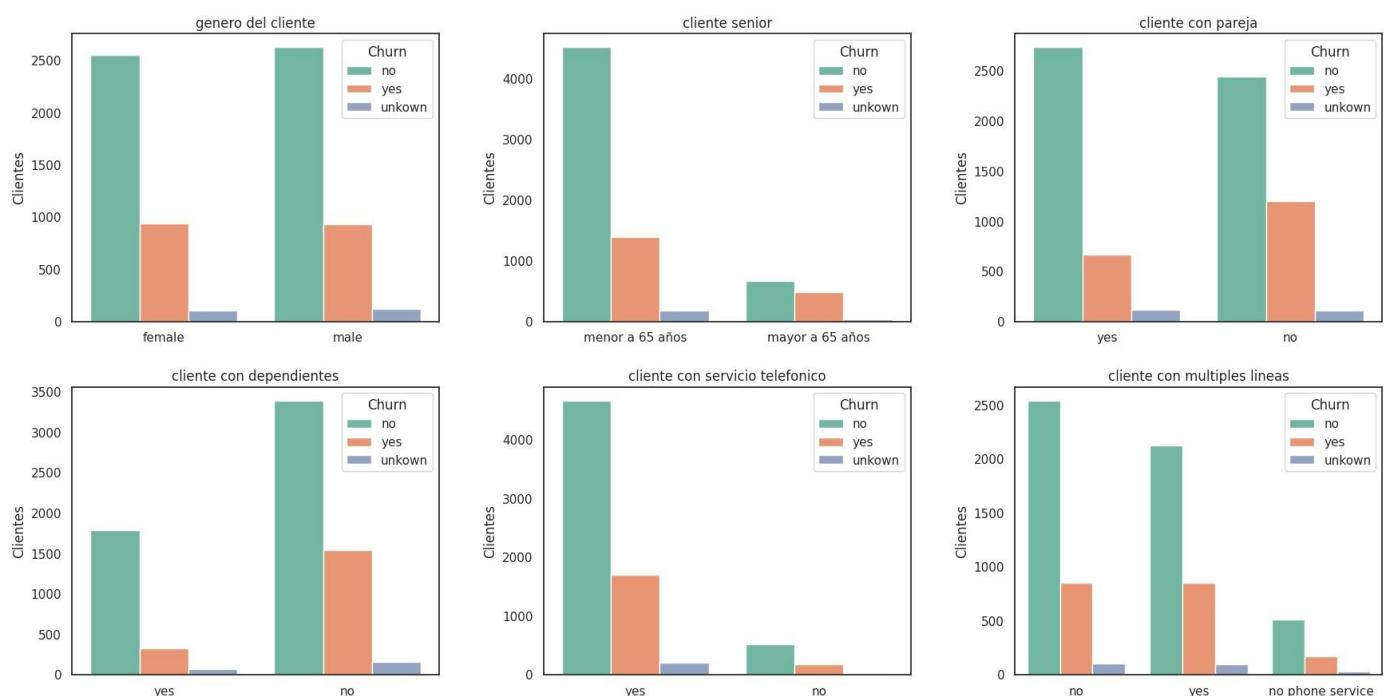
Primero visualizaremos como están conformados nuestra columna de datos Churn que son los clientes que se fueron, se quedaron o en parámetro desconocido



Aquí vemos que la cantidad de clientes que dejaron la empresa siendo está el 25.71% (1869) del total de clientes que tenemos, a comparación de los clientes que no dejaron la empresa siendo el 71.19% (5174) además teniendo con categoría desconocida siendo apenas el 3.1%(224)

Análisis por categorías

Distribución de Churn por variables categóricas



J.G.C

Al analizar estas primeras 6 graficas categóricas vemos que las categorías donde tenemos un mayor numero de clientes de cancelación son:

Clientes sinior: donde vemos que el mayor número de cancelación es por parte de usuarios jóvenes

Clientes con dependientes: donde vemos que el mayor numero de clientes con dependientes que abandonaron el servicio son aquellos que no tienen dependientes bajo su cuidado

Clientes con servicio telefónico: la mayoría que abandonado a si do los que tienen líneas de servicio telefónico siendo este posiblemente algo general combinándose con otros servicios, pero también dandonos información que el servicio sea una posible causa de la deserción del cliente, esto lo podemos confirmar con la retroalimentación de los clientes de la condición del servicio.

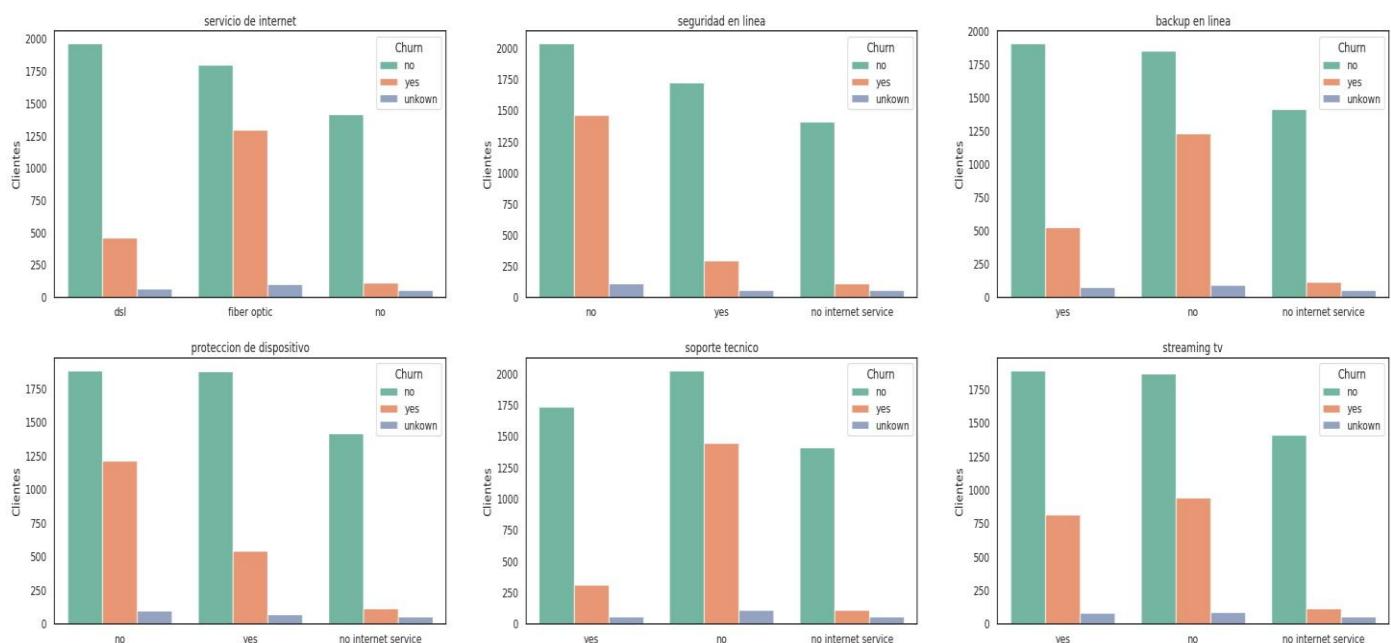
Clientes con pareja: Parece que los clientes sin pareja son más propensos a irse, aunque esto también puede deberse a que hay más clientes solteros en general, esto solo nos da información para marketing para atraer este publico soltero

Clientes con múltiples líneas: tienen un mayor índice de abandonos a diferencia de los que no tienen más de una línea o no tienen ninguna, esto podría deberse a fallas en el servicio u otros factores, es recomendable revisar la retroalimentación del cliente y ver que factor describe mas

Genero del cliente: no observamos variación ni diferencias factibles por lo tanto el género o orientación no es un factor que determine el abandono del servicio, esta información la podemos ignorar

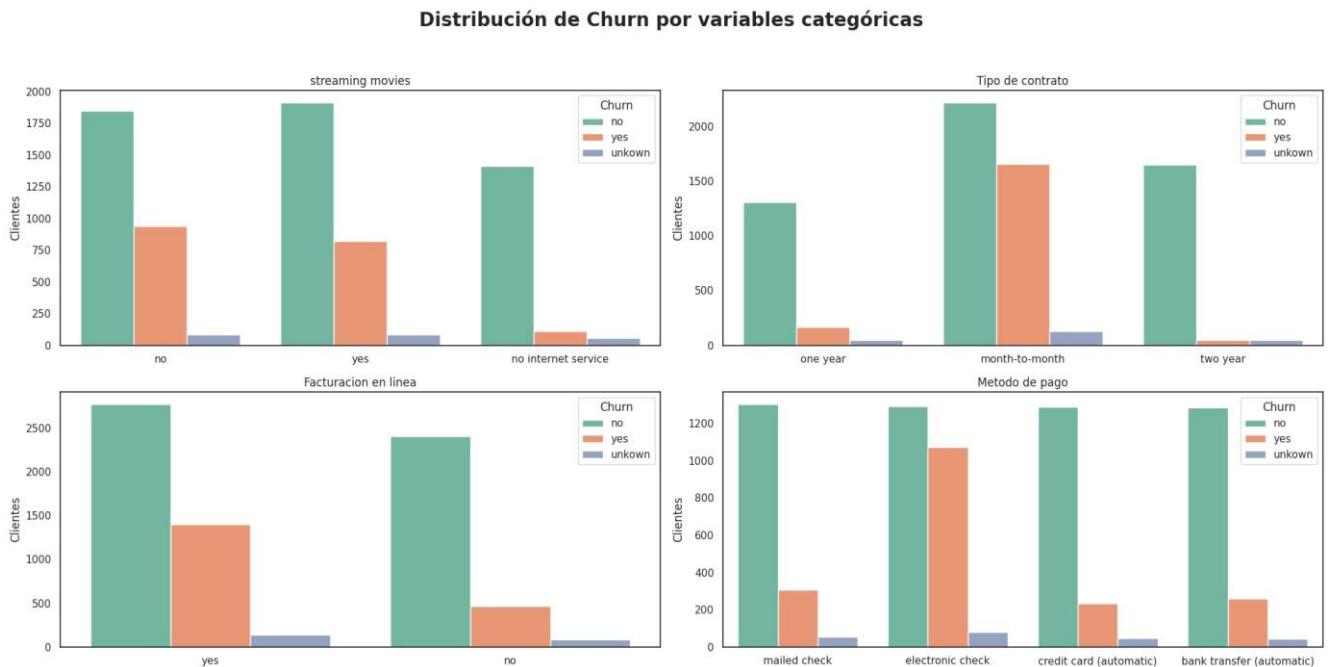
Con el siguiente grupo categórico:

Distribución de Churn por variables categóricas



J.G.C

Podemos visualizar en este conjunto de gráficos que los clientes que mayor abandonan por estas categorías son:



Servicio de internet: los clientes que tienen fibra óptica son el que tiene le mayor numero de abandonos con respecto a este servicio, puede deberse a fallas con respecto a la fibra

Seguridad en línea: aquí nos muestra que los clientes que mas han abandonado han sido los que no tienen servicio de seguridad por lo tanto aquí el servicio se encuentra ofreciendo un buen trabajo y sería recomendable enfocarse principalmente en aumentar las ventas y resolver la pequeña deserción de los que si han dejado el servicio teniendo este sistema de seguridad

Back up en línea: vemos que en esta categoría han abandonado los que no cuentan con este servicio siendo este mas alto que los que si cuentan y no cuentan con internet, recomendable enfocarse en los que han abandonado el servicio y el por qué lo han hecho para mejorarlo y posterior enfocar la venta a un mayor número de clientes para así funcione como una nueva forma de retención de clientes.

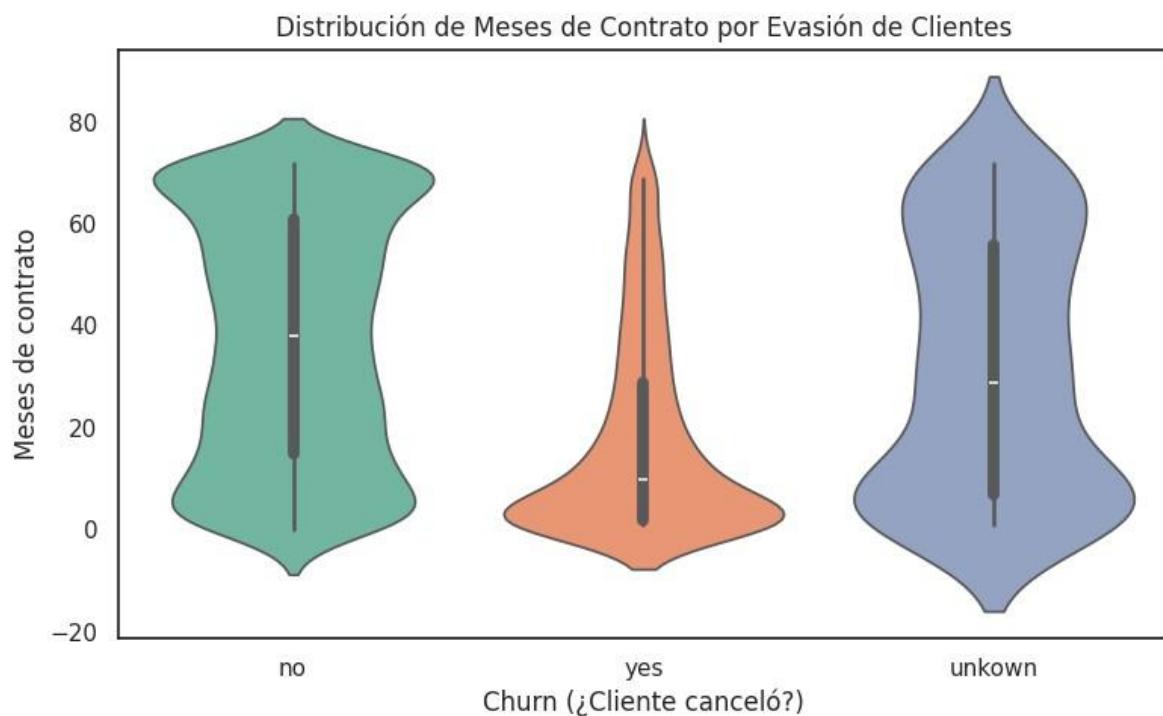
Soporte técnico: aquí observamos que los clientes que abandonaron son lo que no se acercan al servicio atención por lo tanto es importante mejorar y averiguar por qué no confían en el servicio técnico, así como mejorar la atención a este por lo que se fueron y si tuvieron contacto con el servicio

Streming movies: en este aspecto abandonan los que no tuvieron contacto con el servicio de streming, para esto es mejor que nos concentremos en dos factores por que los que tuvieron el servicio abandonaron y el segundo punto como aumentamos las suscripciones de este servicio, pero haciendo énfasis en los que abandonaron antes de pensar en vender más de este servicio

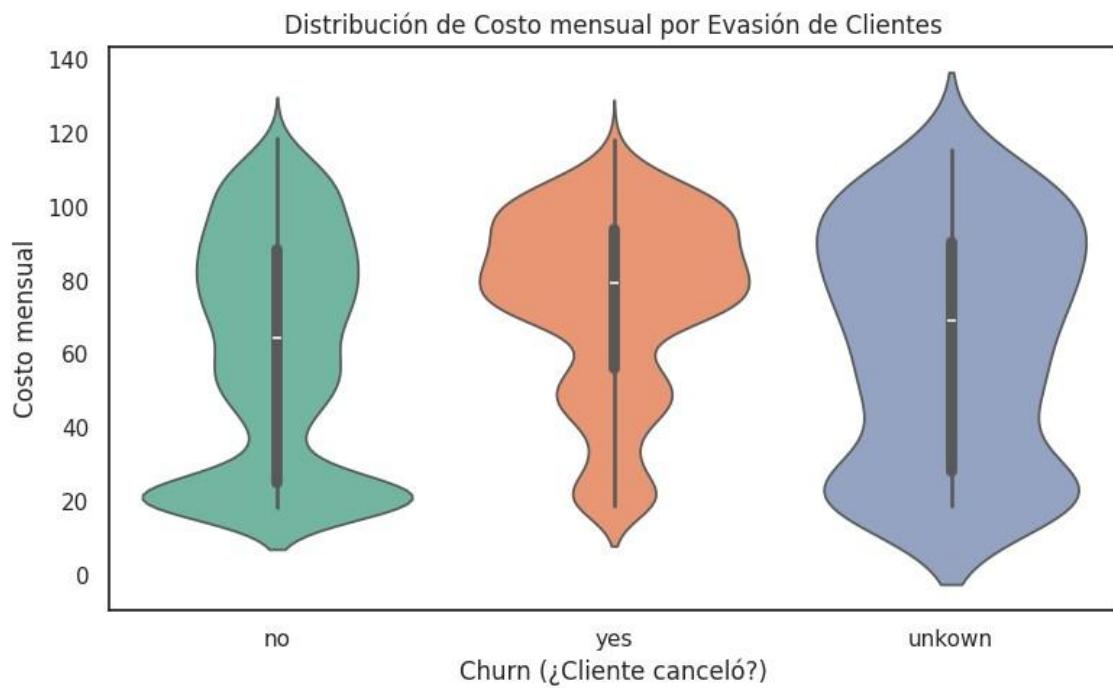
J.G.C

Tipo de contrato: tenemos un mayor numero de abandono con mes a mes esto debe derivarse a que a veces tenemos usuarios que solo contratan para probar el servicio y lo cancelan el siguiente mes o usuarios por motivos económico deciden solo contratar por periodos de tiempo, lo recomendable es verificar las encuestas de calidad acerca de la duración del tiempo del contrato y filtrar si es por prueba o por disgusto

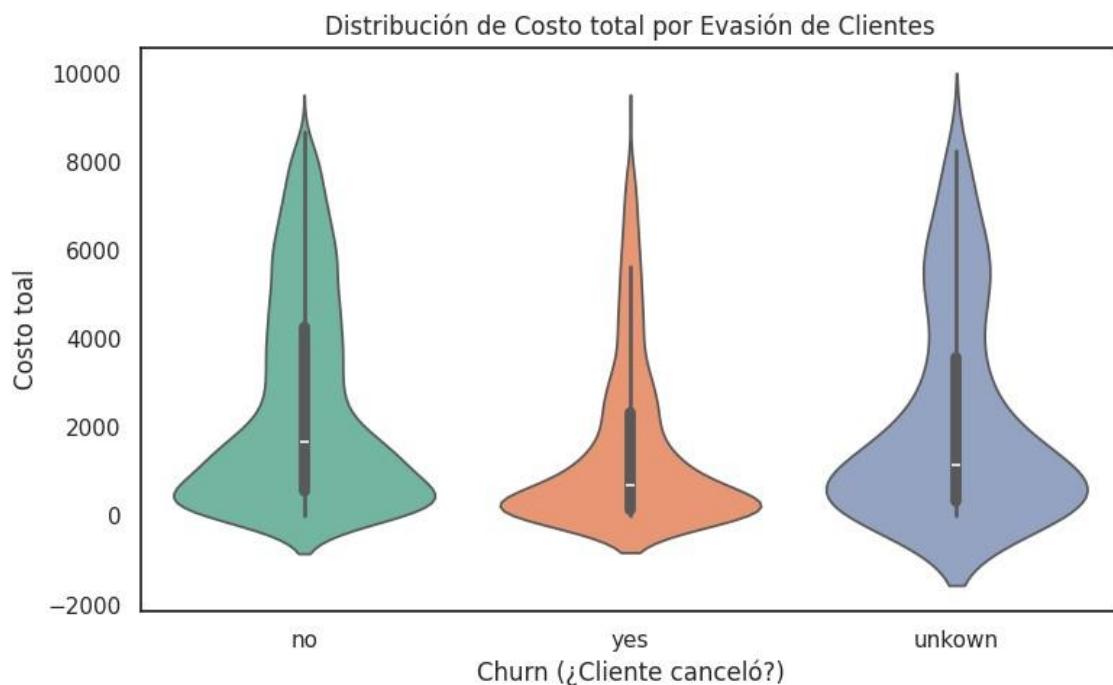
Análisis por categorías económicas



El gráfico de violín muestra claramente que los clientes que abandonan (Churn = yes) tienden a hacerlo en los primeros meses del contrato, con una moda cercana a los 5 meses y una mediana alrededor de 10 meses. En cambio, los clientes que permanecen (Churn = no) suelen tener contratos más largos, con una mediana superior a 40 meses y mayor densidad de retención a largo plazo. Esto sugiere que la fidelización en los primeros meses es clave para evitar la evasión.



Los clientes que abandonaron ("yes") tienen costos mensuales significativamente más altos en promedio que los que no abandonaron ("no"). La distribución de costos para los que abandonaron está desplazada hacia la derecha, indicando valores más altos, la mayoría de los costos, la mayoría de los costos se repiten alrededor de las 80 unidades monetarias, como recomendación los altos costos podrían relacionarse con la razón del abandono (clientes insatisfechos con el servicio con relación al precio), por lo tanto, sería útil investigar que servicios o características están asociados a estos costos elevados que llegan al abandono

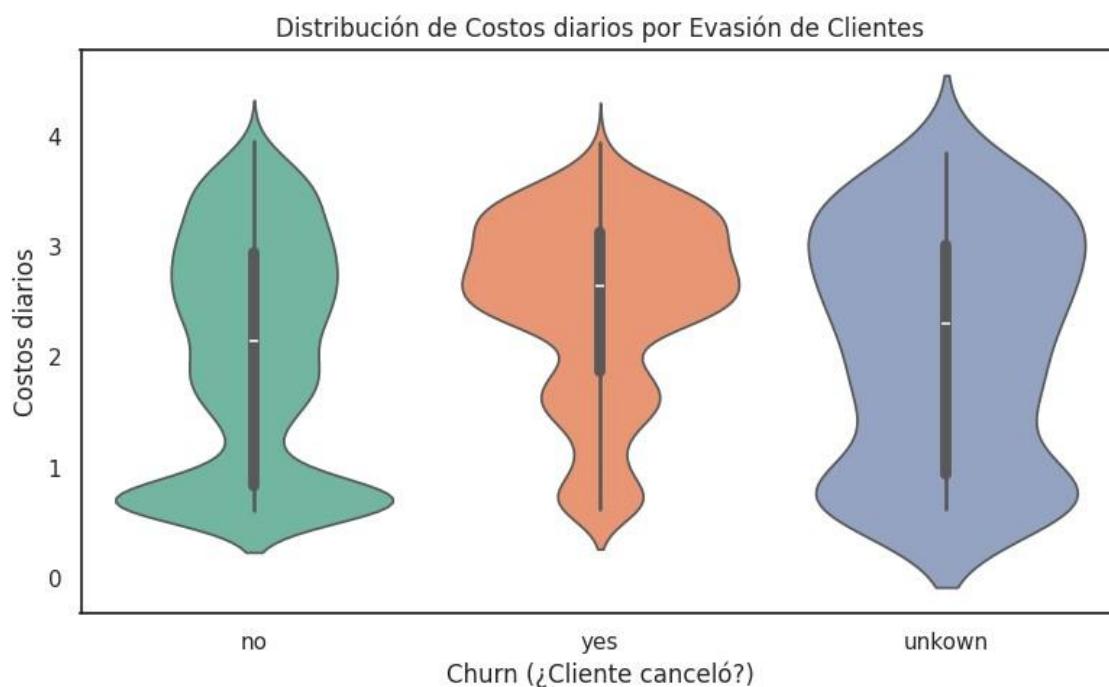


De acuerdo con la gráfica de distribución de costo total por evasión de clientes, al enfocarnos en el parámetro de los clientes si abandonaron vemos que la distribución se

J.G.C

encuentra entre las 6,000 y 10,000 unidades monetarias, donde a entender que tenemos pérdidas financieras al irse los clientes derivado posiblemente a deudas pendientes, multas por cancelación anticipada, costos de recuperación. Esto puede derivar a los altos costos que podrían originar insatisfacción del cliente, teniendo como consecuencia las cancelaciones previas.

Se recomienda averiguar por qué su evasión es tan costosa: ¿Son cargos injustos? ¿Falta de claridad en contratos?, crear un plan de retención temprana para clientes con patrones de quejas o pagos atrasados, así mismo algo que se puede hacer para todos los grupos es optimizar políticas de cobro evitando multas excesivas que impulsen el abandono y una transparencia financiera al comunicar claramente los costos asociados al servicio o cancelación.



Los clientes que cancelaron ("yes") presentan **costos diarios por evasión notablemente más altos** (entre 2 y 3 unidades monetarias) en comparación con los que permanecieron ("no", con costos cercanos a 0). Esto sugiere que su abandono está asociado a **gastos recurrentes elevados**, posiblemente por tarifas diarias percibidas como injustas, falta de flexibilidad en planes, experiencia pobre en la relación costo, derivado de estas posibles variables se recomienda ajustar los cargos diarios que afectan a este grupo, ofrecer opciones mas flexibles para retener clientas sensibles al costo y contactar a los que cancelan para identificar si el motivo principal fue económico y corregir patrones.

Al reducir los costos diarios percibidos como injusto por los clientes podemos disminuir la evasión de este segmento.

Conclusión y recomendaciones generales

El análisis del *churn* en Telecom X revela que la evasión de clientes está impulsada principalmente por factores económicos y categorías específicas de servicio, con un impacto significativo en la rentabilidad. A continuación, se resumen los hallazgos clave y las acciones prioritarias:

Hallazgos Clave

1. Costo Mensual como Driver Principal:

- Los clientes que abandonaron ("yes") pagan en promedio **costos mensuales más altos** (~80 unidades monetarias) frente a los que permanecen.
- Relación costo-valor percibida:** Los altos precios podrían asociarse a insatisfacción con servicios poco diferenciados o cargos no transparentes.

2. Categorías Críticas con Mayor Churn:

- Tipo de contrato:** Los clientes con planes **mes a mes** son los que más cancelan (posiblemente por falta de flexibilidad o atractivo en contratos largos).
- Servicio de internet:** La **fibra óptica** registra alta evasión, sugiriendo fallas técnicas o expectativas no cumplidas.
- Clientes sin dependientes/pareja:** Segmentos con menor compromiso financiero tienden a irse más fácilmente.

3. Patrón Temporal:

- El *churn* se concentra en los **primeros 10 meses**, indicando que la retención inicial es crítica.

4. Costos Diarios y Totales por Evasión:

- Los clientes que se van generan **pérdidas elevadas** (6,000–10,000 unidades), vinculadas a multas, deudas pendientes o costos de recuperación.

Recomendaciones Estratégicas

1. Optimización de Costos y Transparencia:

- Revisar estructura de precios:** Ajustar tarifas de servicios con alta evasión (ej: fibra óptica) y ofrecer paquetes más competitivos.
- Eliminar cargos ocultos:** Comunicar claramente todos los costos al inicio del contrato y en facturas.
- Reducir penalizaciones por cancelación:** Evaluar si las multas están impulsando el abandono.

J.G.C

2. Retención en Categorías Críticas:

- **Contratos:** Incentivar planes anuales con beneficios exclusivos (ej: descuentos, soporte prioritario).
- **Fibra óptica:** Mejorar calidad del servicio y realizar encuestas de satisfacción para identificar fallas técnicas.
- **Clientes solteros/sin dependientes:** Crear campañas de fidelización con ofertas personalizadas (ej: promociones para usuarios individuales).

3. Acciones Proactivas en los Primeros Meses:

- **Programa de onboarding:** Contactar a clientes nuevos en los primeros 3 meses para resolver dudas y ofrecer asistencia.
- **Alertas tempranas:** Monitorear clientes con pagos atrasados o quejas recurrentes para intervenir antes de que cancelen.

4. Recolección de Feedback:

- **Encuestas de salida:** Preguntar a los clientes que cancelan el motivo exacto (ej: costo, servicio, atención).
- **Análisis de datos en tiempo real:** Usar herramientas de analytics para identificar tendencias de *churn* y actuar rápidamente.

Conclusión final

Telecom x puede reducir la evasión de sus clientes al abordar dos puntos principales:

- 1.La percepción de valor económico.
- 2.La experiencia en servicios críticos (fibra óptica, contratos flexibles)

Implementando estas recomendaciones como un punto de partida para así disminuir la evasión y así mejorar la satisfacción del cliente y la sostenibilidad financiera a largo plazo

Nota: Como recomendación final y último paso se recomienda implantar estas acciones en programas piloto con determinado tiempo para evaluar los resultados y posterior crear un análisis final comparando con este reporte, para así comprobar las mejoras fueron efectivas y de alto valor para la resolución de esta problemática.