1c764da3-c2df-4fcc-8500-5c40b0034f48-Copy1

November 20, 2024

1 SPRINT 13 Model Fitness (pronósticos y predicciones)

2 Indice

2.1 Introducción

En el competitivo panorama del fitness, la retención de clientes es un desafío constante. Model Fitness, al igual que muchas otras cadenas de gimnasios, busca optimizar sus estrategias para reducir la tasa de cancelación y aumentar la fidelización de sus usuarios. Ante la necesidad de tomar decisiones basadas en datos sólidos, se ha planteado la presente investigación con el objetivo de desarrollar un modelo predictivo capaz de identificar a los clientes con mayor probabilidad de abandonar el gimnasio.

Este estudio se centra en el análisis de un conjunto de datos que incluye información detallada sobre las características de los usuarios, su historial de uso y su comportamiento de compra. A través de técnicas de aprendizaje automático, como la clasificación y el clustering, se busca identificar patrones y tendencias que permitan predecir la cancelación de la membresía.

2.2 Análisis exploratorio de datos

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import linkage,dendrogram
from sklearn.cluster import KMeans
```

[2]:
```
model_fitness = pd.read_csv("/datasets/gym_churn_us.csv")
```

[3]:
```
model_fitness.columns = model_fitness.columns.str.lower()
model_fitness.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 14 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   gender                             4000 non-null   int64
 1   near_location                      4000 non-null   int64
 2   partner                            4000 non-null   int64
 3   promo_friends                      4000 non-null   int64
 4   phone                              4000 non-null   int64
 5   contract_period                    4000 non-null   int64
 6   group_visits                       4000 non-null   int64
 7   age                                4000 non-null   int64
 8   avg_additional_charges_total       4000 non-null   float64
 9   month_to_end_contract              4000 non-null   float64
 10  lifetime                           4000 non-null   int64
 11  avg_class_frequency_total          4000 non-null   float64
 12  avg_class_frequency_current_month  4000 non-null   float64
 13  churn                              4000 non-null   int64
dtypes: float64(4), int64(10)
memory usage: 437.6 KB
```

[4]:
```
model_fitness.isnull().sum()
```

[4]:
```
gender                             0
near_location                      0
partner                            0
promo_friends                      0
phone                              0
contract_period                    0
group_visits                       0
age                                0
avg_additional_charges_total       0
month_to_end_contract              0
lifetime                           0
avg_class_frequency_total          0
avg_class_frequency_current_month  0
churn                              0
dtype: int64
```

[5]:
```
model_fitness.describe()
```

[5]:

	gender	near_location	partner	promo_friends	phone
count	4000.000000	4000.000000	4000.000000	4000.000000	4000.000000
mean	0.510250	0.845250	0.486750	0.308500	0.903500
std	0.499957	0.361711	0.499887	0.461932	0.295313
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000	0.000000	1.000000
50%	1.000000	1.000000	0.000000	0.000000	1.000000
75%	1.000000	1.000000	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000

	contract_period	group_visits	age
count	4000.000000	4000.000000	4000.000000
mean	4.681250	0.412250	29.184250
std	4.549706	0.492301	3.258367
min	1.000000	0.000000	18.000000
25%	1.000000	0.000000	27.000000
50%	1.000000	0.000000	29.000000
75%	6.000000	1.000000	31.000000
max	12.000000	1.000000	41.000000

	avg_additional_charges_total	month_to_end_contract	lifetime
count	4000.000000	4000.000000	4000.000000
mean	146.943728	4.322750	3.724750
std	96.355602	4.191297	3.749267
min	0.148205	1.000000	0.000000
25%	68.868830	1.000000	1.000000
50%	136.220159	1.000000	3.000000
75%	210.949625	6.000000	5.000000
max	552.590740	12.000000	31.000000

	avg_class_frequency_total	avg_class_frequency_current_month
count	4000.000000	4000.000000
mean	1.879020	1.767052
std	0.972245	1.052906
min	0.000000	0.000000
25%	1.180875	0.963003
50%	1.832768	1.719574
75%	2.536078	2.510336
max	6.023668	6.146783

	churn
count	4000.000000
mean	0.265250
std	0.441521
min	0.000000
25%	0.000000
50%	0.000000

```
75%         1.000000
max         1.000000
```

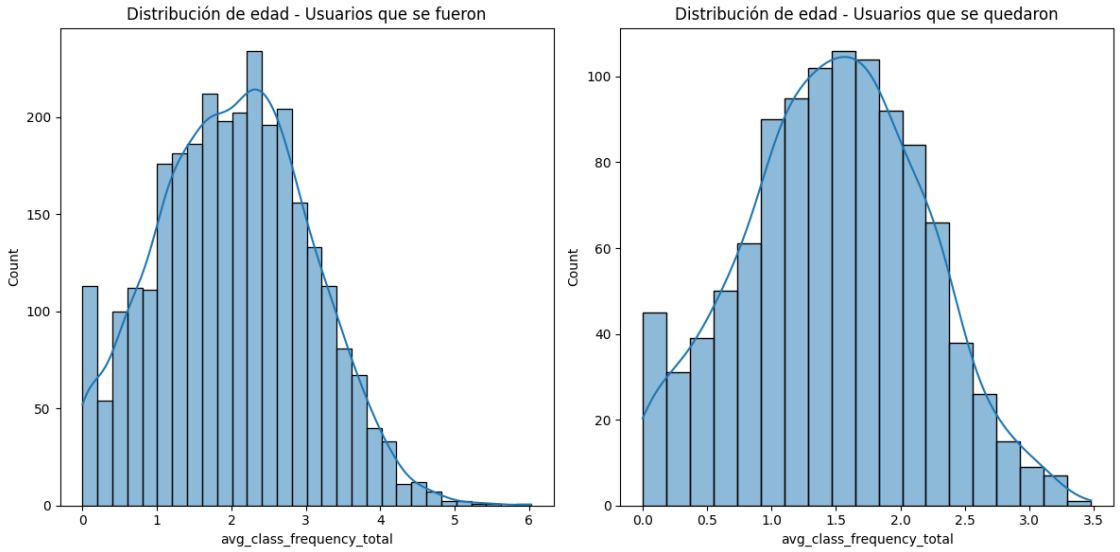
La alta desviación estándar en "contract_period" sugiere que hay una gran diversidad en los tipos de contratos que ofrecen, lo cual podría influir en la tasa de churn.

La edad promedio de los usuarios parece estar en un rango típico para usuarios de gimnasios. Sin embargo, la desviación estándar sugiere que hay una buena representación de diferentes grupos de edad.

Los clientes con contratos más largos (valores altos en 'month_to_end_contract' y 'lifetime') tienden a tener una menor tasa de churn. Esto sugiere que los contratos a largo plazo fomentan la lealtad de los clientes y viceversa.

Los clientes que asisten con mayor frecuencia a las clases (valores altos en 'avg_class_frequency_total' y 'avg_class_frequency_current_month') tienen una menor tasa de churn. Esto indica que la participación activa en las actividades del gimnasio aumenta la satisfacción del cliente y reduce la probabilidad de cancelación.

[6]:
```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 6))

# Histograma para los usuarios que se fueron (churn=0)
sns.histplot(data=model_fitness[model_fitness['churn'] == 0],␣
 ↪x='avg_class_frequency_total', kde=True, ax=axes[0])
axes[0].set_title('Distribución de edad - Usuarios que se fueron')

# Histograma para los usuarios que se quedaron (churn=1)
sns.histplot(data=model_fitness[model_fitness['churn'] == 1],␣
 ↪x='avg_class_frequency_total', kde=True, ax=axes[1])
axes[1].set_title('Distribución de edad - Usuarios que se quedaron')

plt.tight_layout()
plt.show()
```



Los resultados sugieren que existe una fuerte relación entre la frecuencia de clases y la probabilidad de que un cliente cancele su membresía. Los usuarios que asisten al gimnasio con mayor frecuencia tienen una mayor probabilidad de permanecer como miembros.

La distribución bimodal de los usuarios que se fueron indica que puede ser necesario segmentar a estos clientes en dos grupos diferentes: aquellos que cancelan debido a una baja utilización del gimnasio y aquellos que cancelan por otras razones.

Se pueden elaborar estrategia para reducir la tasa de churn, es importante enfocarse en aumentar la frecuencia de visitas de los clientes, especialmente aquellos que tienen una baja frecuencia de clases. Esto puede lograrse mediante la implementación de programas de incentivos, la personalización de las recomendaciones de clases y la mejora de la experiencia del usuario.

[7]:
```
grupos_cancelacion = model_fitness.groupby('churn')
promedio = grupos_cancelacion.mean()
promedio
```

[7]:

churn	gender	near_location	partner	promo_friends	phone	contract_period	group_visits	age	avg_additional_charges_total	month_to_end_contract	lifetime	avg_class_frequency_total	avg_class_frequency_current_month
0	0.510037	0.873086	0.534195	0.353522	0.903709	5.747193	0.464103	29.976523	158.445715	5.283089	4.711807	2.024876	2.027882
1	0.510839	0.768143	0.355325	0.183789	0.902922	1.728558	0.268615	26.989632	115.082899	1.662582	0.990575	1.474995	1.044546

Se pueden observar algunas tendencias interesantes:

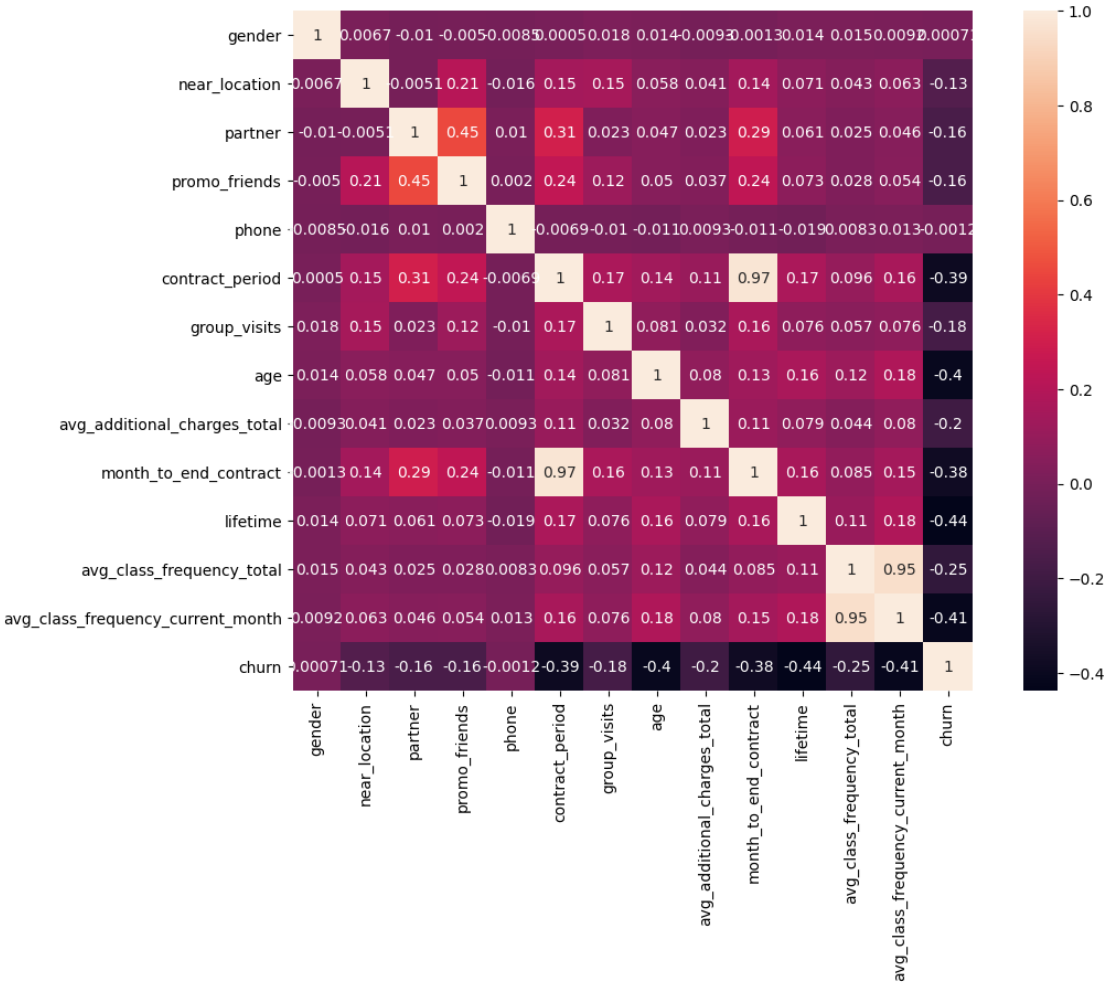
Ubicación: Los usuarios que se quedaron tienen una ligeramente mayor proporción de personas que viven cerca del gimnasio.

Relación: Los usuarios que se quedaron tienen una mayor proporción de personas con pareja y que fueron recomendados por amigos.

Contrato: Los usuarios que se quedaron tienen contratos de mayor duración y, por lo tanto, más meses restantes hasta la finalización del contrato.

[8]:
```
cm = model_fitness.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(cm, annot = True, square=True)
```

[8]: <AxesSubplot:>



"avg_class_frequency_total" y "avg_class_frequency_current_month" tienen una correlación muy alta (cercana a 1), lo que sugiere que la frecuencia promedio de visitas en general está muy relacionada con la frecuencia en el mes actual. Esto es esperable, ya que los usuarios que suelen ir al gimnasio con frecuencia tienden a mantener ese hábito.

"contract_period" y "month_to_end_contract" están fuertemente correlacionadas. Esto se debe a que a mayor duración del contrato, más meses quedan hasta su finalización.

"churn" y "contract_period" tienen una correlación negativa esto sugiere que los usuarios con contratos más largos tienen menos probabilidades de cancelar su membresía.

Por ultimo, "churn" y variables relacionadas con la frecuencia de visitas indica que los usuarios que

van al gimnasio con más frecuencia tienen menos probabilidades de cancelar.

2.3 Construcción de modelo

[9]:
```
X = model_fitness.drop('churn', axis=1)
y = model_fitness['churn']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[28]:
```
# Regresión logística
modelo_rl = LogisticRegression()
modelo_rl.fit(X_train, y_train)

# Bosque aleatorio
modelo_ba = RandomForestClassifier()
modelo_ba.fit(X_train, y_train)

# Hacer predicciones en el conjunto de prueba
y_pred_modelo_rl = modelo_rl.predict(X_test)
y_pred_modelo_ba = modelo_ba.predict(X_test)

print("Regresión Logística:")
print(confusion_matrix(y_test, y_pred_modelo_rl))
print(classification_report(y_test, y_pred_modelo_rl))
print("\n----------------------------------------------------")
print("Bosque Aleatorio:")
print(confusion_matrix(y_test, y_pred_modelo_ba))
print(classification_report(y_test, y_pred_modelo_ba))
```

```
/opt/conda/envs/python3/lib/python3.9/site-
packages/sklearn/linear_model/_logistic.py:763: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

```
Regresión Logística:
[[573  25]
 [ 47 155]]
              precision    recall  f1-score   support

           0       0.92      0.96      0.94       598
```

```
           1       0.86      0.77      0.81       202

    accuracy                           0.91       800
   macro avg       0.89      0.86      0.88       800
weighted avg       0.91      0.91      0.91       800


------------------------------------------------
Bosque Aleatorio:
[[574  24]
 [ 47 155]]
              precision    recall  f1-score   support

           0       0.92      0.96      0.94       598
           1       0.87      0.77      0.81       202

    accuracy                           0.91       800
   macro avg       0.90      0.86      0.88       800
weighted avg       0.91      0.91      0.91       800
```

Datos interesantes de resaltar de los resultados obtenidos:

*Matriz de confusión del modelo Regresion logica: El modelo predijo que 573 muestras eran de la clase 0 de Verdaderos positivos y 155 muestras eran de la clase 1 de Verdaderos negativos.

Ambos modelos tienen un buen desempeño general los valores de accuracy, precisión, recall y F1-score son altos tanto para la regresión logística como para el bosque aleatorio.

El bosque aleatorio parece tener un ligero mejor desempeño, en general en las métricas, especialmente en la precisión.

El número de falsos negativos (47) es ligeramente superior en la regresión logística comparado con el bosque aleatorio (40). Esto significa que la regresión logística tiende a clasificar más casos positivos como negativos.

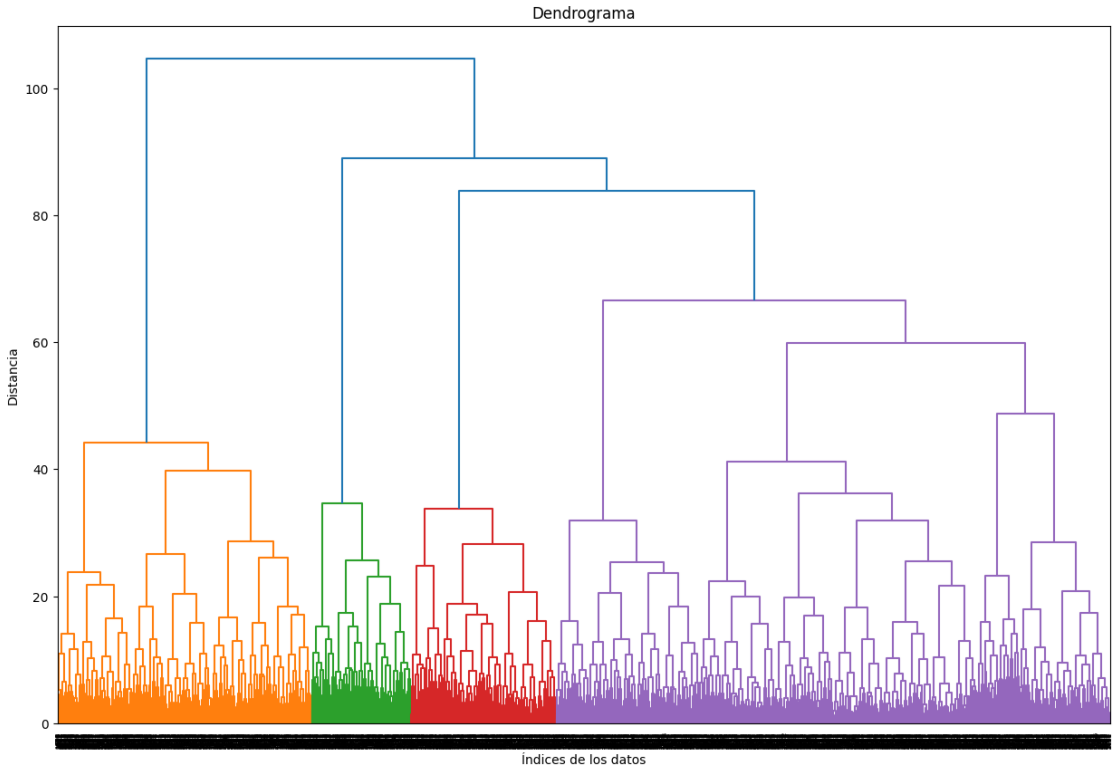
El modelo de bosque aleatorio parece ser un modelo más preciso y equilibrado, con un buen desempeño tanto en precisión como en recall. sin embargo, la regresión logística también es un buen modelo, pero podría ser menos preciso en identificar todos los casos positivos.

2.4 Clústeres de usuarios

[11]:
```
# Estandarizacion de datos

scaler = StandardScaler()
X_train_st = scaler.fit_transform(X_train)
X_test_st = scaler.transform(X_test)
```

[12]:
```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Matriz de distancias y el dendrograma
Z = linkage(X_scaled, 'ward')
plt.figure(figsize=(15, 10))
dendrogram(Z)
plt.title('Dendrograma')
plt.xlabel('Índices de los datos')

plt.ylabel('Distancia')
plt.show()
```



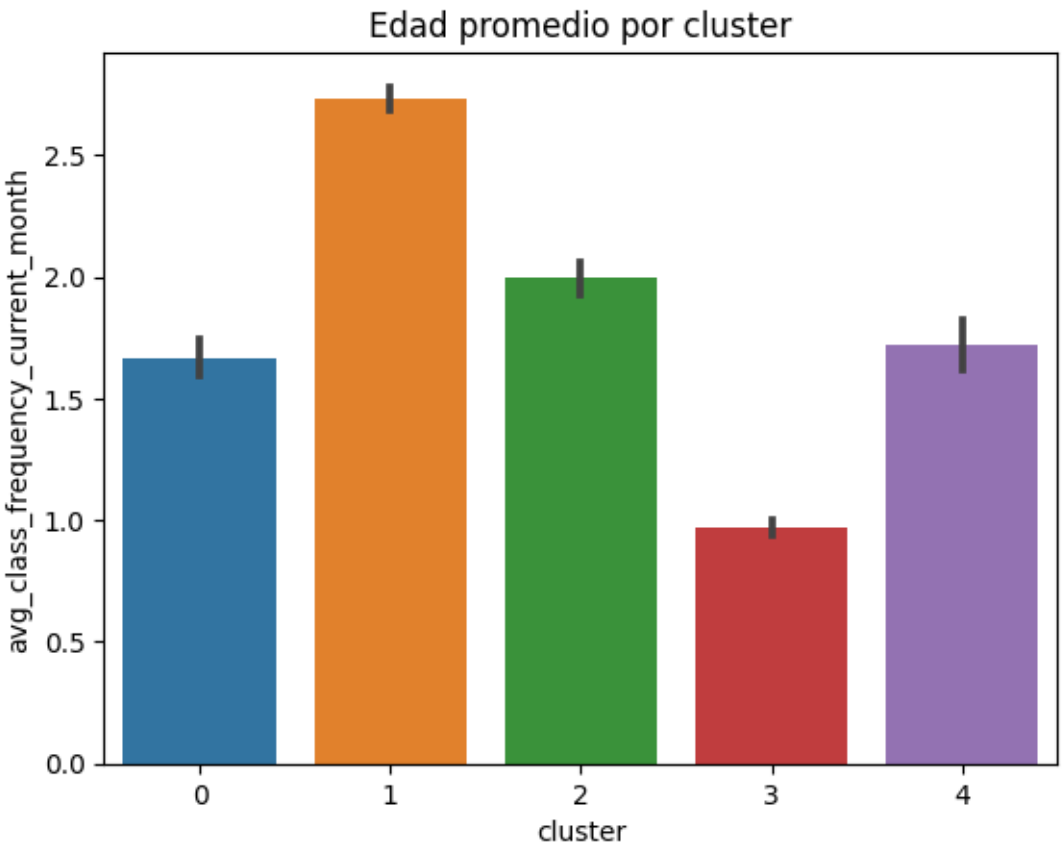
El número óptimo de clústeres sugerido serian 4, se puede ver por la division de colores

[27]:
```
# Modelo K-means
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(X_scaled)
y_kmeans = kmeans.predict(X_scaled)

model_fitness['cluster'] = y_kmeans
print(model_fitness.head())
```

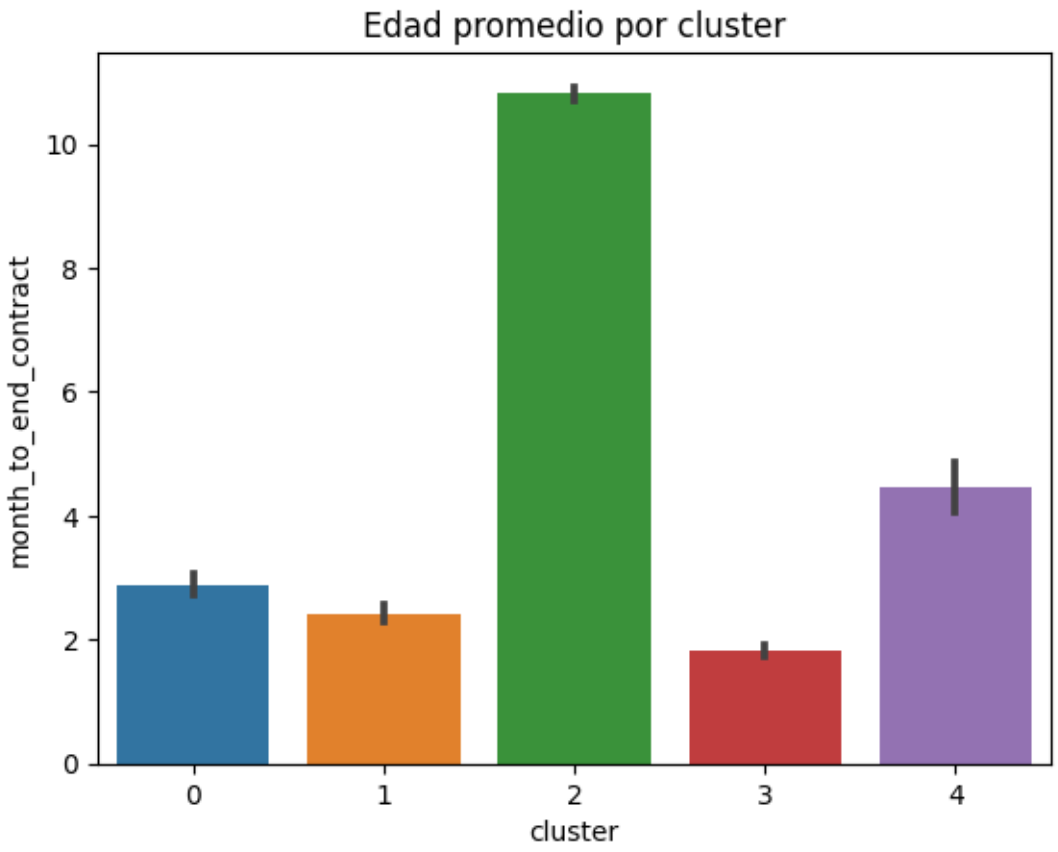
```
   gender  near_location  partner  promo_friends  phone  contract_period  \
0       1              1        1              1      0                6
```

```
1       0              1        0              0      1                12
2       0              1        1              0      1                 1
3       0              1        1              1      1                12
4       1              1        1              1      1                 1

   group_visits  age  avg_additional_charges_total  month_to_end_contract  \
0             1   29                     14.227470                    5.0
1             1   31                    113.202938                   12.0
2             0   28                    129.448479                    1.0
3             1   33                     62.669863                   12.0
4             0   26                    198.362265                    1.0

   lifetime  avg_class_frequency_total  avg_class_frequency_current_month  \
0         3                   0.020398                           0.000000
1         7                   1.922936                           1.910244
2         2                   1.859098                           1.736502
3         2                   3.205633                           3.357215
4         3                   1.113884                           1.120078

   churn  cluster
0      0        4
1      0        2
2      0        3
3      0        2
4      0        0
```

[26]:
```
cluster_means = model_fitness.groupby('cluster').mean()
cluster_std = model_fitness.groupby('cluster').std()

# Visualizar las medias
sns.barplot(x='cluster', y='avg_class_frequency_current_month',␣
 ↪data=model_fitness)
plt.title('Edad promedio por cluster')
plt.show()
```



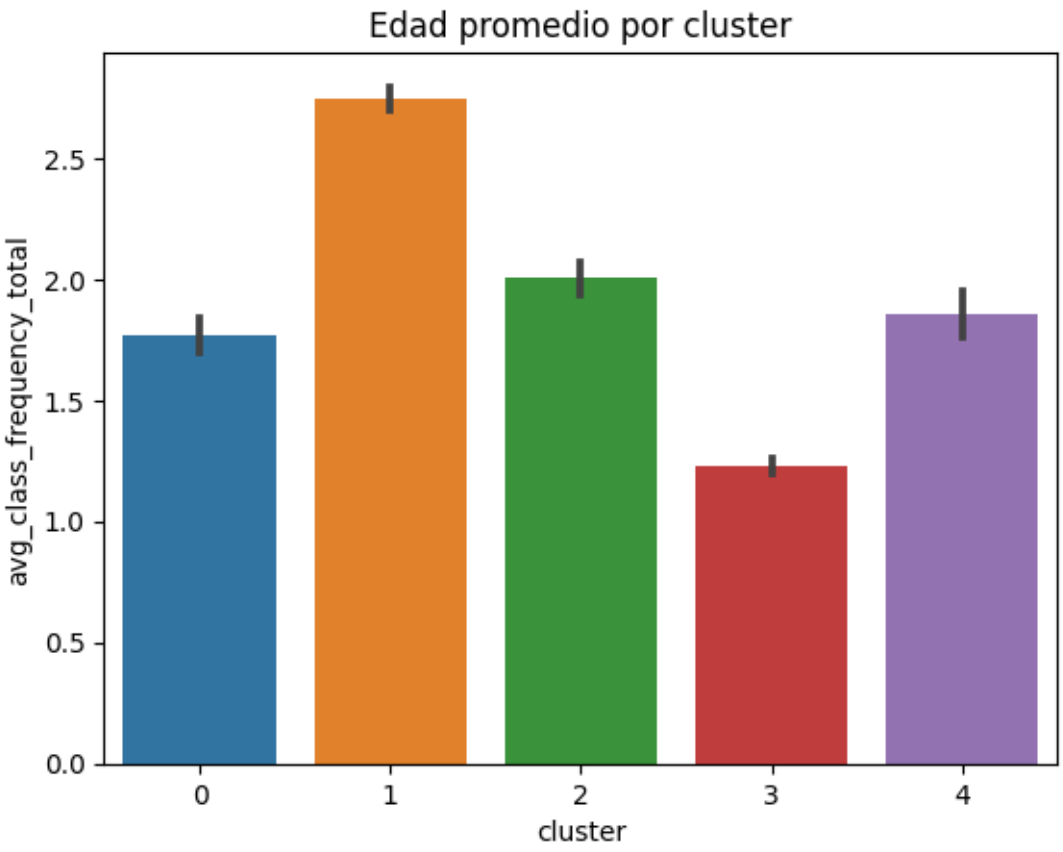
[25]:
```
cluster_means = model_fitness.groupby('cluster').mean()
cluster_std = model_fitness.groupby('cluster').std()

# Visualizar las medias
sns.barplot(x='cluster', y='month_to_end_contract', data=model_fitness)
plt.title('Edad promedio por cluster')
plt.show()
```



El análisis de la duración de los contratos revela una clara segmentación de los clientes. El cluster 2 se destaca por tener contratos significativamente más largos, lo que sugiere un mayor compromiso con el gimnasio. Por otro lado, los otros clusters presentan contratos más cortos, lo que podría indicar una mayor volatilidad en su base de clientes."

La duración del contrato parece ser un factor clave para predecir el churn. Los clientes con contratos más largos tienen una menor probabilidad de cancelar su membresía. Esta información puede ser utilizada para desarrollar estrategias de retención personalizadas para los clientes con contratos más cortos

[19]:
```
cluster_means = model_fitness.groupby('cluster').mean()
cluster_std = model_fitness.groupby('cluster').std()

# Visualizar las medias
sns.barplot(x='cluster', y='avg_class_frequency_total', data=model_fitness)
plt.title('Edad promedio por cluster')
plt.show()
```



El análisis de clustering reveló una clara segmentación de los usuarios en función de su frecuencia promedio de visitas. Observamos que el cluster 1 presenta la frecuencia más alta, mientras que el cluster 3 muestra la más baja. Esta variabilidad sugiere que existen diferentes perfiles de uso dentro de la base de clientes.

[22]:
```
# Calcular la tasa de churn por cluster
churn_rate_by_cluster = model_fitness.groupby('cluster')['churn'].mean()
print(churn_rate_by_cluster)
```

```
cluster
0    0.246445
1    0.089989
2    0.021965
3    0.572942
4    0.266839
Name: churn, dtype: float64
```

Al analizar la relación entre la frecuencia de visitas y la tasa de churn, encontramos una correlación inversa significativa. El cluster 1, con la frecuencia de visitas más baja, presenta la tasa de churn más alta. Por el contrario, el cluster 2, con la frecuencia de visitas más alta, muestra la tasa de churn más baja. Estos resultados sugieren que la frecuencia de visitas es un factor determinante

en la probabilidad de que un cliente cancele su membresía

2.5 Conclusiones y recomendaciones

*Conclusiones:

La frecuencia de visitas es el principal factor de retención, es decir, los clientes que asisten con más frecuencia al gimnasio tienen una mayor probabilidad de permanecer como miembros.Tambien los clientes valoran las experiencias personalizadas y las recomendaciones adaptadas a sus necesidades, por eso la segmentación de clientes permite estrategias más efectivas como identificar diferentes perfiles de clientes que permita diseñar programas y ofertas más relevantes.Otro patron a resaltar es que los contratos a largo plazo fomentan la lealtad, los clientes con contratos más largos tienen una menor probabilidad de cancelar.

*Recomendaciones Estratégicas: 1. Fomentar la frecuencia de visitas:

Programas de recompensas: Implementar un sistema de puntos o niveles que premie a los clientes por su frecuencia de visitas. Desafíos y competencias: Organizar desafíos y competencias para motivar a los clientes a alcanzar metas específicas. Clases variadas y atractivas: Ofrecer una amplia gama de clases y actividades para satisfacer los diferentes intereses y niveles de condición física de los clientes.

2. Personalizar la experiencia del cliente:

Recomendaciones personalizadas: Utilizar algoritmos de recomendación para sugerir clases, entrenadores y servicios adicionales basados en los intereses y el historial de cada cliente. Comunicación personalizada: Enviar mensajes personalizados a través de diferentes canales (email, app, SMS) con ofertas y contenido relevante. Planes de entrenamiento personalizados: Ofrecer planes de entrenamiento personalizados diseñados por profesionales del fitness.

3. Fortalecer la comunidad del gimnasio:

Eventos sociales: Organizar eventos sociales y actividades grupales para fomentar la interacción entre los miembros. Redes sociales: Crear una comunidad en línea donde los miembros puedan compartir sus experiencias y logros. Programas de referidos: Incentivar a los clientes a recomendar el gimnasio a sus amigos y familiares.

4. Optimizar los contratos y precios:

Ofertas atractivas: Ofrecer promociones y descuentos especiales para incentivar la contratación de planes a largo plazo. Flexibilidad: Permitir a los clientes personalizar sus planes y adaptar su membresía a sus necesidades cambiantes.

5. Monitorear y mejorar continuamente:

Análisis de datos: Realizar análisis periódicos de los datos para identificar nuevas tendencias y oportunidades de mejora. Encuestas de satisfacción: Realizar encuestas de satisfacción para conocer la opinión de los clientes y detectar áreas de mejora. Pruebas A/B: Implementar pruebas A/B para evaluar la efectividad de diferentes estrategias y optimizar las campañas de marketing.

En resumen, la clave para mejorar la retención de clientes en Model Fitness es centrarse en ofrecer una experiencia personalizada y atractiva que fomente la lealtad y el compromiso a largo plazo. Al combinar estas estrategias con un enfoque basado en datos, Model Fitness podrá aumentar significativamente su tasa de retención y fortalecer su posición en el mercado.