재구매 및 금액 예측 기반 고객 가치 분석 프로젝트

1. 프로젝트 개요

분석의 목표:

- 리테일 회사의 과거 고객 거래 데이터를 기반으로 BG/NBD 모델을 적용하여,

각 고객의 미래 재구매 행동을 예측하고 고객 가치를 세분화하기.

- Gamma-Gamma 모델을 이용하여 고객이 “한 번 구매할 때 얼마를 지출하는지(평균 거래 금액)”를 추정하기.

In [ ]:
```
import pandas as pd
import numpy as np
```

In [ ]:
```
# %pip install lifetimes
```

2. 데이터 준비

- **구글 빅쿼리** 이용해서 필요한 데이터를 분석에 맞게 출력

```
SELECT CustomerID
     , COUNT(DISTINCT DATE(TransactionDate)) - 1 AS frequency
     , DATE_DIFF(DATE(MAX(TransactionDate)), DATE(MIN(TransactionDate)), DAY) AS recency

     # discount된걸 고려함
     , AVG(TotalAmount) AS monetary_value

     # 현재날짜를 2024년 5월 1일이라고 가정
     , DATE_DIFF('2024-05-01', DATE(MIN(TransactionDate)), DAY) AS T
FROM `eminent-ring-451902-n9.ai.retail_transaction`
GROUP BY CustomerID
HAVING frequency > 0
LIMIT 1000;
```

In [ ]:
```
df = pd.read_csv('/content/sample_data/retail_freqmoreone.csv', index_col=0)
df.head()    # frequency = 재구매/추가 구매 횟수  //  recency = 첫구매와 마지막 구매 사이의 기간 // monetary = 평균 구매 금액 (한번의 구매에서
```

Out[ ]:

CustomerID	frequency	recency	monetary_value	T
121413	1	103	448.843328	131
818911	1	79	258.033574	104
418277	1	195	448.615587	349
628270	1	67	241.698151	332
147124	1	335	196.334731	342

- 위의 정보를 토대로, 고객의 앞으로의 얼마나 자주 구매할지(BG/NBD 모델)
- 한번 구매할때 얼마나 쓸지(Gamma-Gamma 모델)등을 예측 가능.

3. BG/NBD 모델

- 이 모델을 이용해 고객이 "앞으로 얼마나 몇번 구매할지/구매를 지속할지"를 예측할것임.
- 각 고객마다 “활성 상태가 끝나는 시점”이 다를 수 있으며, 구매 횟수 분포를 음이항 분포(NBD)로, 이탈(비활성화) 과정을 베타 분포로 가정

In [ ]:
```
data = df[['frequency', 'recency', 'T']]
data.head()
```

Out[ ]:

CustomerID	frequency	recency	T
121413	1	103	131
818911	1	79	104
418277	1	195	349
628270	1	67	332
147124	1	335	342

In [ ]:
```
print(data[['frequency', 'recency', 'T']].describe())
print(data[['frequency', 'recency', 'T']].isnull().sum())
```

```
         frequency      recency            T
count  1000.000000  1000.000000  1000.000000
mean      1.046000   123.748000   249.207000
std       0.218934    85.765122    85.725685
min       1.000000     1.000000    13.000000
25%       1.000000    51.000000   185.000000
50%       1.000000   112.000000   263.000000
75%       1.000000   185.000000   321.250000
max       3.000000   356.000000   367.000000
frequency    0
recency      0
T            0
dtype: int64
```

In [ ]:
```
from lifetimes import BetaGeoFitter     # BetaGeoFitter는 frequency, recency, T를 입력값으로 삼음.


# BG/NBD 모델 객체 생성
bgf = BetaGeoFitter(penalizer_coef=0.01)

# 데이터에 있는 frequency, recency, T를 이용해 모델 fitting
bgf.fit(data['frequency'], data['recency'], data['T'])

# 모델 파라미터 출력
bgf
```

Out[ ]:
```
<lifetimes.BetaGeoFitter: fitted with 1000 subjects, a: 0.66, alpha: 229.96, b: 0.06, r: 2.26>
```

1. 모델 파라미터 해석:

- **r, alpha**:
  - 구매 빈도를 음이항 분포(Negative Binomial)로 가정했을때 관련된 모수들
  - 고객의 구매 횟수 분포를 결정
- **a, b**:
  - 고객이 이탈(비활성화)할 때까지의 생존 시간을 베타 분포(Beta)로 가정했을때 관련된 모수들
  - 고객이 활성 상태를 유지할 확률을 결정
- 모델이 학습되면, 각 고객이 **향후 얼마나 자주 구매할지**, 고객이 **여전히 활성 상태인지** 등도 계산 가능!

---

2. penalizer_coef(정규화 계수):

- 왜 필요?
  - 데이터가 매우 적거나, 고객별로 극단적인 frequency/recency 분포가 있을때, 모델이 극단적으로 치우칠수있음(overfitting).
  - 0보다 큰값(예: 0.0001, 001)을 설정하면, 파라미터가 너무 커지거나 작아지는것들 방지해서, **모델이 좀 더 안정적**.
- 값의 범위:
  - 일반적으로 0.0 ~ 0.1 사이의 작은값.
  - 0.0 이면 정규화 없이 그대로 피팅.

In [ ]:
```
bgf.summary
```

Out[ ]:

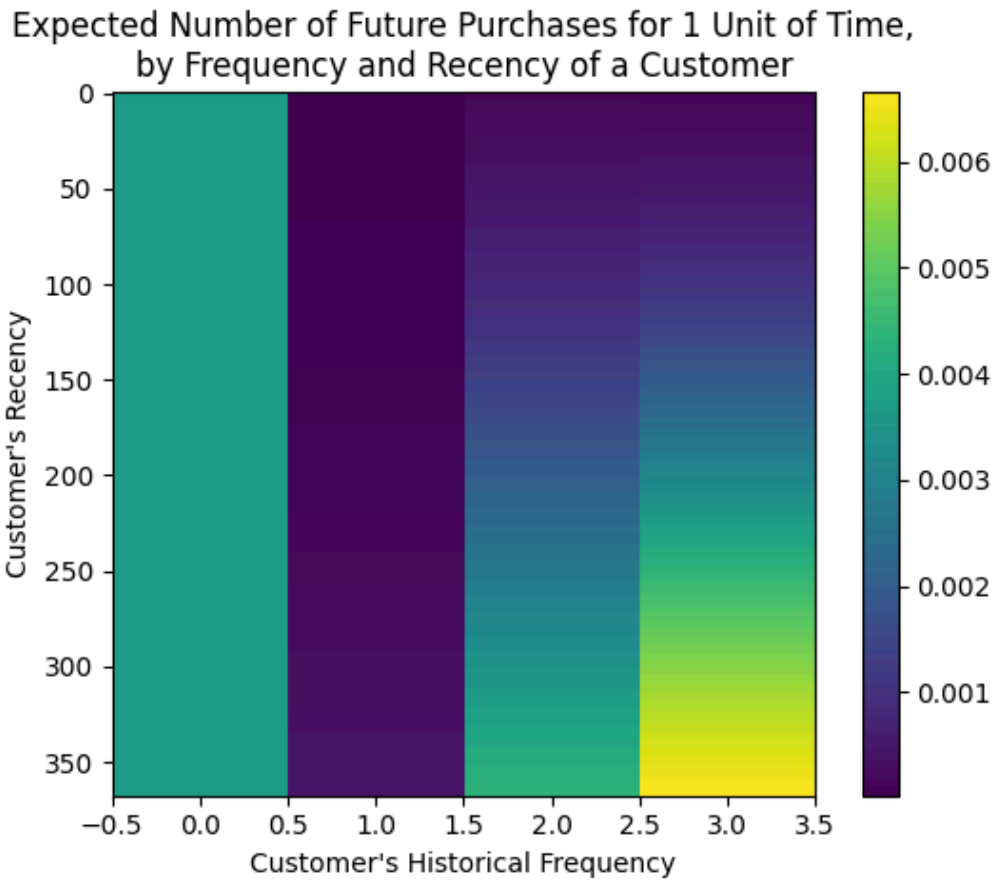
	coef	se(coef)	lower 95% bound	upper 95% bound
r	2.260015	0.125195	2.014633	2.505396
alpha	229.960700	16.886742	196.862685	263.058715
a	0.662256	0.149945	0.368363	0.956149
b	0.055920	0.015328	0.025877	0.085962

BG/NBD 모델로 예측된 미래 구매 횟수의 시각화 (각각의 frequency 축과 recency축에 따라)

- 각 좌표에 대해 "향후 일정 기간(시간단위) 동안 예상 구매 횟수"가 색상으로 표현됨.
- = "앞으로의 구매 횟수 기대치"

In [ ]:
```
from lifetimes.plotting import plot_frequency_recency_matrix

plot_frequency_recency_matrix(bgf)
```

Out[ ]:
```
<Axes: title={'center': 'Expected Number of Future Purchases for 1 Unit of Time,\nby Frequency and Recency of a Custome
r'}, xlabel="Customer's Historical Frequency", ylabel="Customer's Recency">
```



활용 포인트

- 시각적으로 고객 세분화:

어떤 (frequency, recency) 구간에 고객들이 많이 분포하는지 확인하고, 해당 구간에 따라 다른 마케팅 전략을 세울 수 있습니다.

- 모델 직관 파악:

BG/NBD 모델이 “빈도(F)”와 “최근성(R)”을 어떻게 결합해 향후 구매를 예측하는지 한눈에 이해할 수 있습니다.

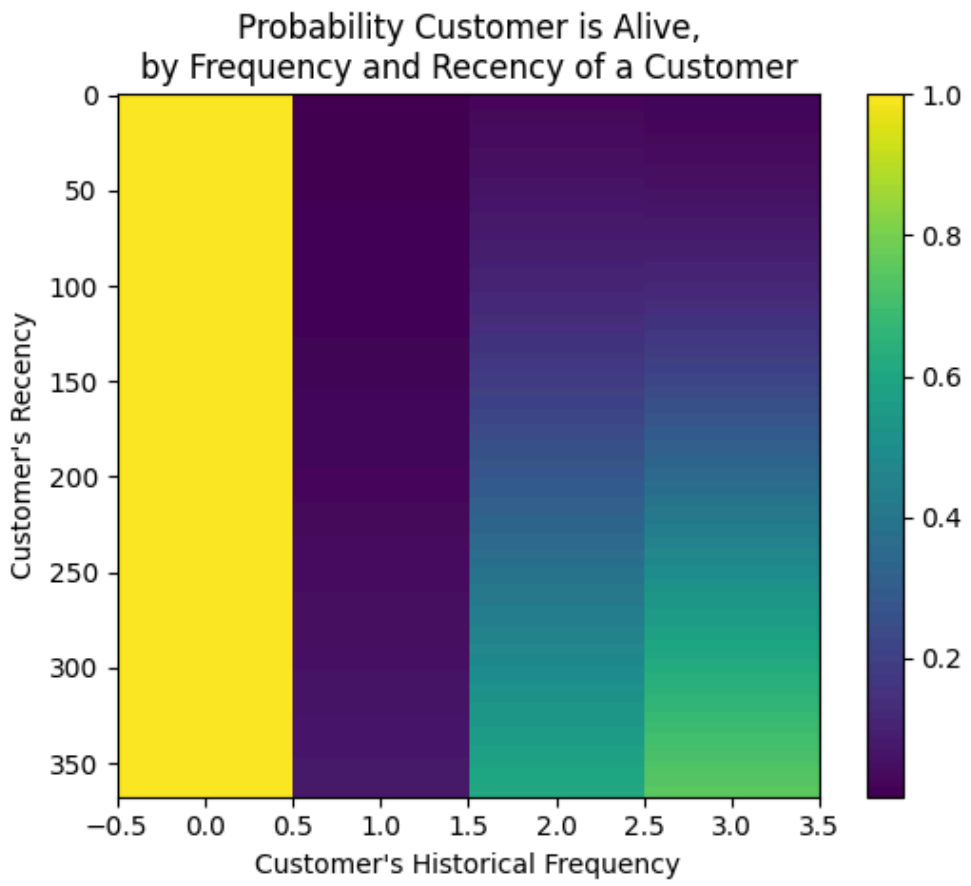
- 추가 분석 지표와 결합:

이 매트릭스 위에 고객들의 실제 분포를 점(Scatter)으로 표시하거나, 구매 금액(monetary)을 색깔/크기로 시각화하면 더 심층적인 인사이트를 얻을 수 있습니다.

BG/NBD 모델로 추정한 고객이 아직 활성(Active) 상태일 확률의 시각화

In [ ]:
```
from lifetimes.plotting import plot_probability_alive_matrix

plot_probability_alive_matrix(bgf)
```

Out[ ]:
```
<Axes: title={'center': 'Probability Customer is Alive,\nby Frequency and Recency of a Customer'}, xlabel="Customer's His
torical Frequency", ylabel="Customer's Recency">
```



활용 포인트

- 잠재적 VIP 식별:

Frequency가 높고 Recency도 짧아 Alive Probability가 높은 고객들은 가장 가치 있는 그룹으로, 리텐션/충성도 제고 활동에 집중할 수 있습니다.

- 휴면/이탈 가능 고객 파악:

Frequency가 낮거나 Recency가 길어 Alive Probability가 낮게 나온 고객은 재활성화 캠페인(win-back)을 고려하거나, 이탈 고객으로 분류해 다른 전략을 세울 수 있습니다.

- 마케팅 우선순위 설정:

전체 고객을 (frequency, recency) 기반으로 세분화하여, 각 구간에 맞는 마케팅 자원을 배분할 수 있습니다.

BG/NBD 모델을 이용해 고객들이 향후 일정 기간(t=1 단위) 동안 예상 구매 횟수가 높은 순으로 고객을 랭킹하는 방법

- 핵심 아이디어는 BetaGeoFitter 객체의 메서드인

conditional_expected_number_of_purchases_up_to_time(t, frequency, recency, T) 를 사용해 각 고객별로 **"다음 t 기간 동안 예상 구매 횟수"**를 구한 뒤, 그 값을 기준으로 고객을 정렬하는 것입니다.

In [ ]:
```
# conditional_expected_number_of_purchases_up_to_time(t, frequency, recency, T) 메서드 활용
t = 7  # 앞으로 일주일 동안의 구매 예측
data['predicted_purchases'] = bgf.conditional_expected_number_of_purchases_up_to_time(t, data['frequency'], data['recency'


# 정렬하고, 상위 n 명 출력
data.sort_values(by='predicted_purchases', ascending=False).head(5)  # 여기선 상위 5명
```

Out[ ]:

CustomerID	frequency	recency	T	predicted_purchases
340516	3	204	208	0.062279
294753	2	74	83	0.054933
892820	3	195	242	0.049623
693531	2	136	143	0.047032
818275	2	136	148	0.045346

In [ ]:
```
# 앞으로 30일 동안
t = 30
data['predicted_purchases'] = bgf.conditional_expected_number_of_purchases_up_to_time(t, data['frequency'], data['recency'
```

```
# 정렬하고, 상위 n 명 출력
data.sort_values(by='predicted_purchases', ascending=False).head(5)
```

Out[ ]:

CustomerID	frequency	recency	T	predicted_purchases
340516	3	204	208	0.259558
294753	2	74	83	0.225307
892820	3	195	242	0.207212
693531	2	136	143	0.194194
818275	2	136	148	0.187319

In [ ]:
```
for t in [7, 30, 90]:
    col = f'predicted_purchases_{t}d'
    data[col] = bgf.conditional_expected_number_of_purchases_up_to_time(
        t,
        data['frequency'],
        data['recency'],
        data['T']
    )
```

In [ ]:
```
# 앞으로 90일 동안 예측
t = 90
data['predicted_purchases'] = bgf.conditional_expected_number_of_purchases_up_to_time(t, data['frequency'], data['recency'


# 정렬하고, 상위 n 명 출력
data.sort_values(by='predicted_purchases', ascending=False).head(5)
```

Out[ ]:

CustomerID	frequency	recency	T	predicted_purchases	predicted_purchases_7d	predicted_purchases_30d	predicted_purchases_90d
340516	3	204	208	0.728845	0.062279	0.259558	0.728845
294753	2	74	83	0.611746	0.054933	0.225307	0.611746
892820	3	195	242	0.584289	0.049623	0.207212	0.584289
693531	2	136	143	0.534418	0.047032	0.194194	0.534418
818275	2	136	148	0.515993	0.045346	0.187319	0.515993

1. t = 1:

- 예측할 기간에 따라 t로 설정합니다.
- 데이터셋에서 사용 중인 시간 단위(주/일 등)에 따라 달라집니다(여기서 사용된 데이터셋은 하루 단위).

2. predicted_purchases 열 생성:

- BG/NBD 모델(bgf)의 conditional_expected_number_of_purchases_up_to_time 메서드를 호출합니다.
- 각 고객이 향후 t 기간 내 구매할 것으로 예측되는 횟수를 계산해 predicted_purchases 열에 저장합니다.

세그먼트 기준

세그먼트	기준	의미
VIP	> 0.10	다음 30일 내 재구매가 활발하게 예상되는 최상위 고객
Potential	0.05 ~ 0.10	적당한 재구매 가능성 있음
Passive	0.01 ~ 0.05	가능성은 낮지만 살아있는 고객
Churn Risk	≤ 0.01	이탈 가능성이 매우 높은 고객

In [ ]:
```
# 간단한 기준으로 등급 분류
def classify_customer(x):
    if x > 0.10:
        return 'VIP'
    elif x > 0.05:
        return 'Potential'
    elif x > 0.01:
        return 'Passive'
    else:
        return 'Churn Risk'
```

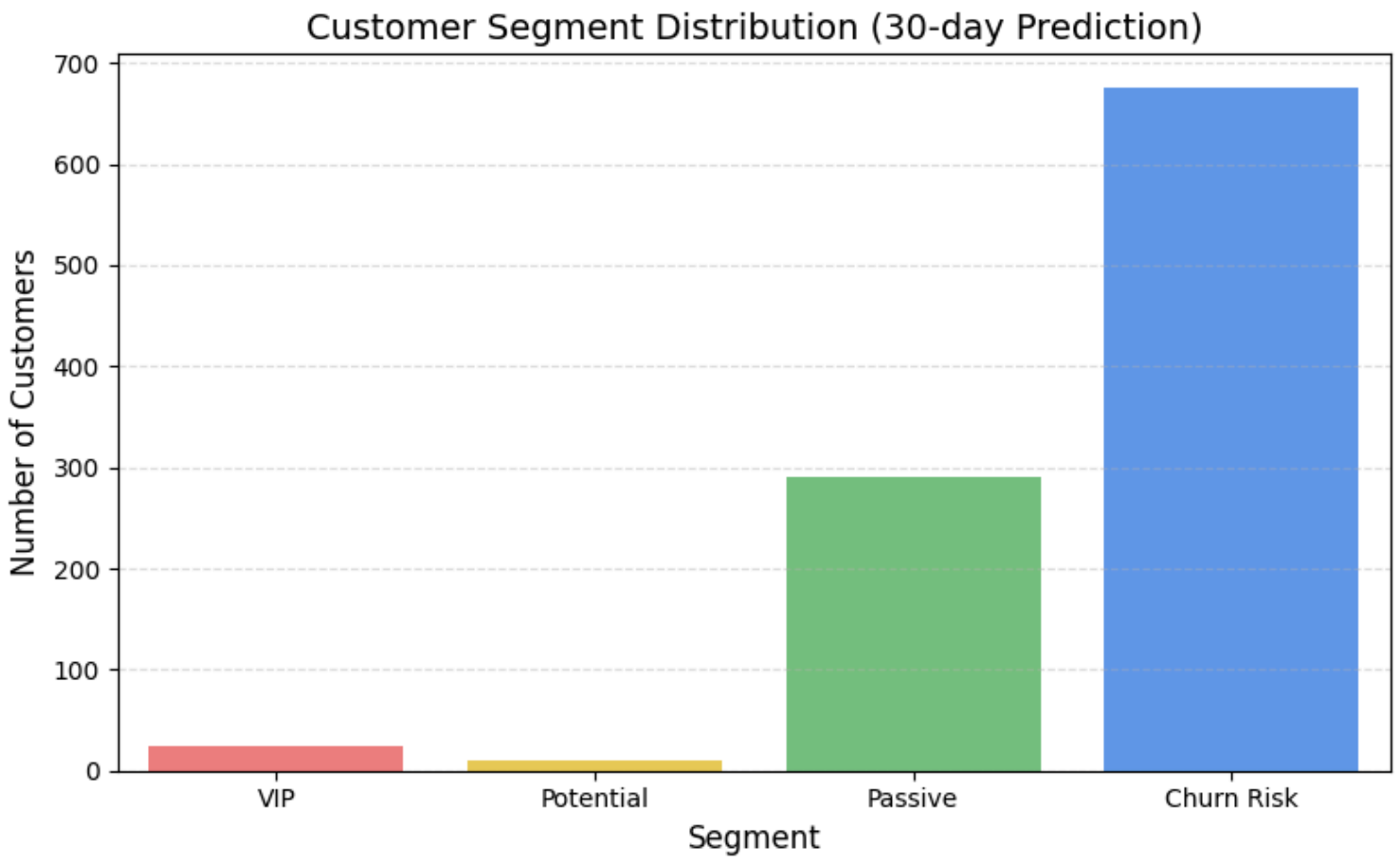
```
data['segment'] = data['predicted_purchases_30d'].apply(classify_customer)

# 분포 확인
data['segment'].value_counts()
```

Out[ ]:

segment	count
Churn Risk	675
Passive	291
VIP	24
Potential	10

dtype: int64

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# 세그먼트별 색상 지정
segment_palette = {
    'VIP': '#FF6B6B',         # Red
    'Potential': '#FFD93D',   # Yellow
    'Passive': '#6BCB77',     # Green
    'Churn Risk': '#4D96FF'   # Blue
}

# 시각화
plt.figure(figsize=(8, 5))
sns.countplot(data=data, x='segment', order=['VIP', 'Potential', 'Passive', 'Churn Risk'], palette=segment_palette)

plt.title('Customer Segment Distribution (30-day Prediction)', fontsize=14)
plt.xlabel('Segment', fontsize=12)
plt.ylabel('Number of Customers', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()
```



결과:

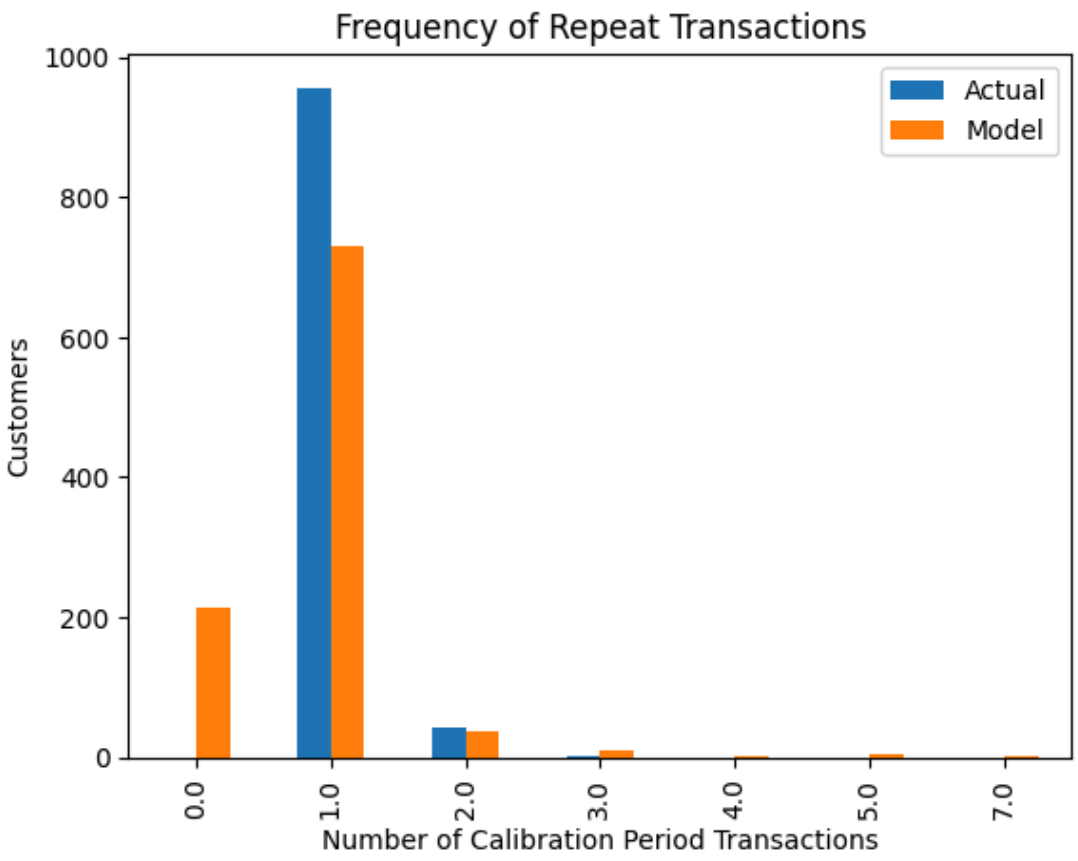
- 전체 고객 중 다수는 향후 재구매 가능성이 낮은 "Passive", "Churn Risk"로 분류됨. (리테일 데이터의 특성인듯, 대부분의 고객은 1~2번만 구매하고 다시 돌아오지 않음.)
- 상위 10% 고객은 재구매 가능성이 높고, 예측 구매 횟수도 높음. → **이들은 충성도 강화 및 리텐션 프로그램의 우선 대상.**
- 일부 Passive 고객은 최근 활동은 적지만 여전히 살아 있을 확률이 높음. → **재참여 캠페인으로 전환 가능성 존재**

BG/NBD 모델의 적합도(fit)를 시각적으로 평가하는 방법 (2가지를 비교함)

1. 실제 데이터에서 관측된 재구매 분포
2. 모델로부터 시뮬레이션된 재구매 횟수 분포

In [ ]:
```
from lifetimes.plotting import plot_period_transactions
plot_period_transactions(bgf)
```

Out[ ]:
```
<Axes: title={'center': 'Frequency of Repeat Transactions'}, xlabel='Number of Calibration Period Transactions', ylabel
='Customers'>
```



- calibration(= 모델을 학습할때 사용한 기간)

- 가로축 => 고객이 calibration기간동안 재구매를 몇번했는지에 따라 구간으로 나눔.
- 세로축 => 해당 재구매 횟수에 속한 고객의 수

해석

- 두 분포 거의 일치 = 모델이 실제 데이터 잘 설명!
- 두 분포 크게 어긋남 = 모델이 잘 추정하지 못함. (데이터 전처리나 pernalizer_coef같은 파라미터 재조정 필요)

개별 고객 단위로 예측과 확률 변화를 구체적으로 확인/조회하는 방법

- 이전 내용과 핵심 모델(BG/NBD)을 사용하는 원리는 같지만
- 여기에선 위에 학습한 모델을 활용해 개별 고객에 맞게 "predict" 하는거

In [ ]:
```
t = 10 # 30일간의 예측

individual = data.iloc[20]
bgf.predict(t, individual['frequency'], individual['recency'], individual['T'])
```

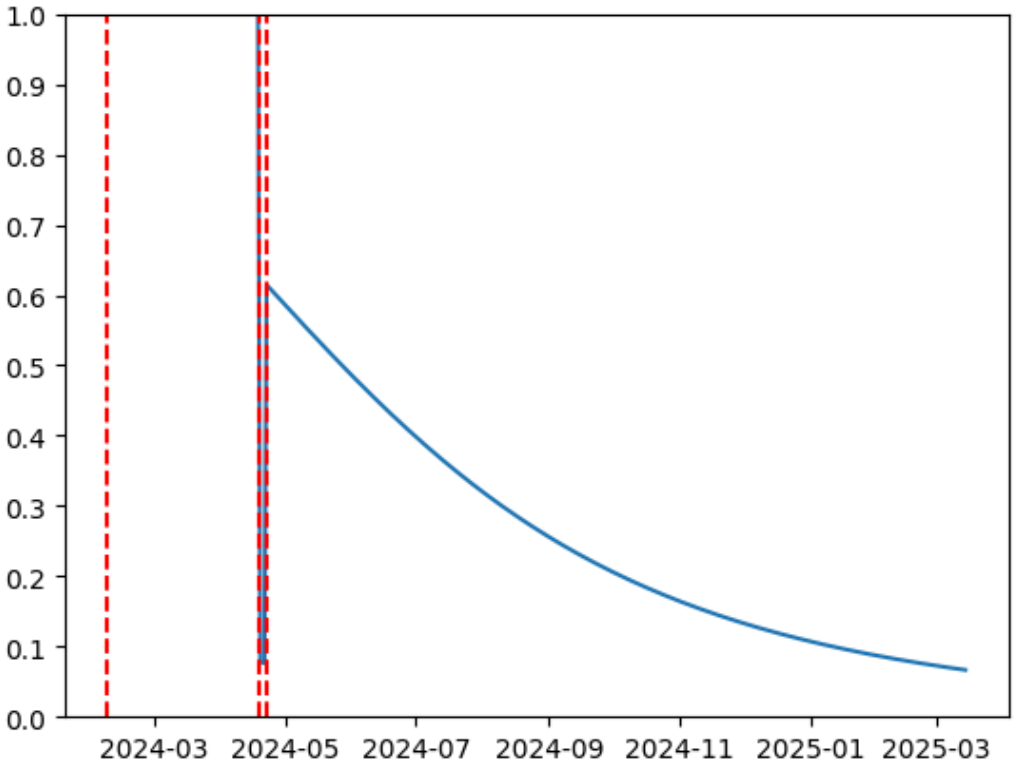
Out[ ]:
```
np.float64(0.005807447546305386)
```

In [ ]:
```
transaction_data = pd.read_csv('/content/sample_data/Retail_Transaction_Dataset.csv')
transaction_data.head()
```

Out[ ]:

	CustomerID	ProductID	Quantity	Price	TransactionDate	PaymentMethod	StoreLocation	ProductCategory	DiscountApplied(%)
0	109318	C	7	80.079844	12/26/2023 12:32	Cash	176 Andrew Cliffs\nBaileyfort, HI 93354	Books	18.677100
1	993229	C	4	75.195229	8/5/2023 0:00	Cash	11635 William Well Suite 809\nEast Kara, MT 19483	Home Decor	14.121365
2	579675	A	8	31.528816	3/11/2024 18:51	Cash	910 Mendez Ville Suite 909\nPort Lauraland, MO...	Books	15.943701
3	799826	D	5	98.880218	10/27/2023 22:00	PayPal	87522 Sharon Corners Suite 500\nLake Tammy, MO...	Books	6.686337
4	121413	A	7	93.188512	12/22/2023 11:38	Cash	0070 Michelle Island Suite 143\nHoland, VA 80142	Electronics	4.030096

In [ ]:
```
from lifetimes.plotting import plot_history_alive


id = 294753
days_since_birth = 400

sp_trans = transaction_data.loc[transaction_data['CustomerID'] == id]
plot_history_alive(bgf, days_since_birth, sp_trans, 'TransactionDate')
```



4. Gamma-Gamma 모델이란?

- 목적:
  - 고객이 “한 번 구매할 때 얼마를 지출하는지(평균 거래 금액)”를 추정
  - BG/NBD 모델이 “얼마나 자주 구매할 것인가?”에 초점을 맞췄다면, Gamma-Gamma 모델은 “구매 금액 규모”를 예측하는 역할을 합니다.
- 전제:
  - 고객별 구매 금액이 독립적이며, (대체로) Gamma 분포를 따른다고 가정
  - BG/NBD에서 Frequency와 Monetary가 독립이라는 가정(혹은 상관관계가 낮다는 전제)이 필요합니다.

In [ ]:
```
# monetary_value까지 있는 데이터셋 불러오기
# 최소 한번의 재구매를 한 사람들만 (재방문 고객들 대상!)
df.head()
```

Out[ ]:

CustomerID	frequency	recency	monetary_value	T
121413	1	103	448.843328	131
818911	1	79	258.033574	104
418277	1	195	448.615587	349
628270	1	67	241.698151	332
147124	1	335	196.334731	342

중요 포인트: Gamma-Gamma 모델에서 CLV를 계산할때, 'monetary value'와 'purchase frequency' 사이에 관련성은 없다는 전제가 있어야 함.

- 그래서 현실에선, 이 모델을 사용하기 위해선 이 두 가지 사이에 Pearson Correlation이 0에 가까운지 확인해야함.

In [ ]:
```
# 재방문 고객들 사이에서 'monetary_value'와 'frequency'의 연관성
df[['monetary_value', 'frequency']].corr()
```

Out[ ]:

	monetary_value	frequency
monetary_value	1.000000	0.040852
frequency	0.040852	1.000000

In [ ]:
```
# 연관성을 확인했으면, 모델 피팅
from lifetimes import GammaGammaFitter

ggf = GammaGammaFitter(penalizer_coef = 0.01)
ggf.fit(df['frequency'],                # *여기서 학습시킨건 재방문고객들 데이터만임
        df['monetary_value'])
ggf
```

Out[ ]: <lifetimes.GammaGammaFitter: fitted with 1000 subjects, p: 4.02, q: 0.38, v: 3.80>

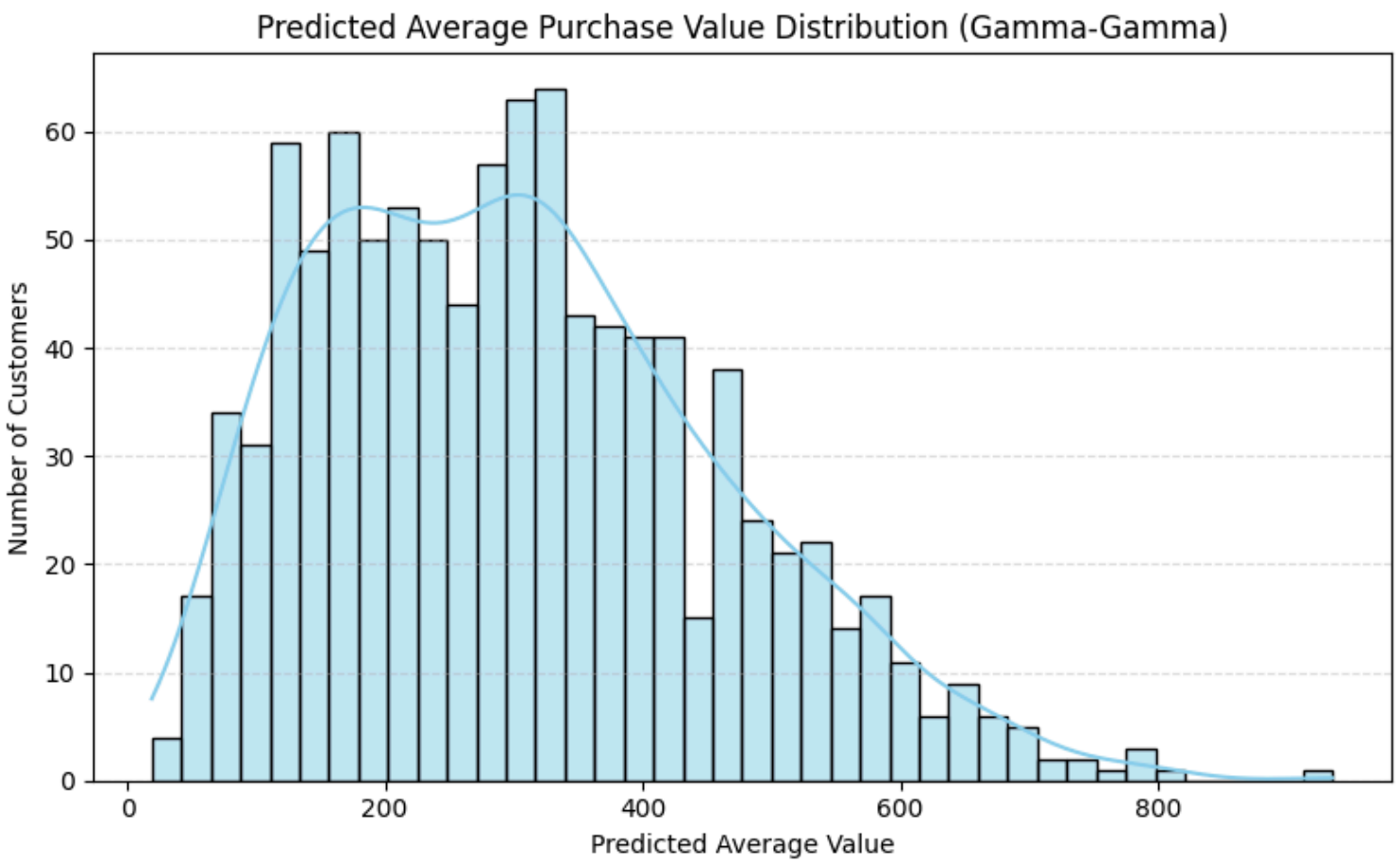
In [ ]:
```
# ggf가 제공하는 메서드로 "각 고객이 향후 한번 구매할때 기대되는 평균 구매 금액"을 예측
result = ggf.conditional_expected_average_profit(
        df['frequency'],                  # *근데 여기선 전체 고객들(재방문 안한사람들도 포함)을 대상으로 예측
        df['monetary_value']
    )         # 각 고객ID별로 예측된 평균 구매금액이 시리즈 형태로

result.head(10)
```

Out[ ]:

CustomerID	0
121413	534.960998
818911	309.449982
418277	534.691839
628270	290.143747
147124	236.530392
589251	161.260519
608105	433.630972
201514	228.668338
91439	277.374586
301389	510.046635

dtype: float64

In [ ]:
```
plt.figure(figsize=(8, 5))
sns.histplot(result, bins=40, kde=True, color='skyblue')
plt.title('Predicted Average Purchase Value Distribution (Gamma-Gamma)')
plt.xlabel('Predicted Average Value')
plt.ylabel('Number of Customers')
plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()
```

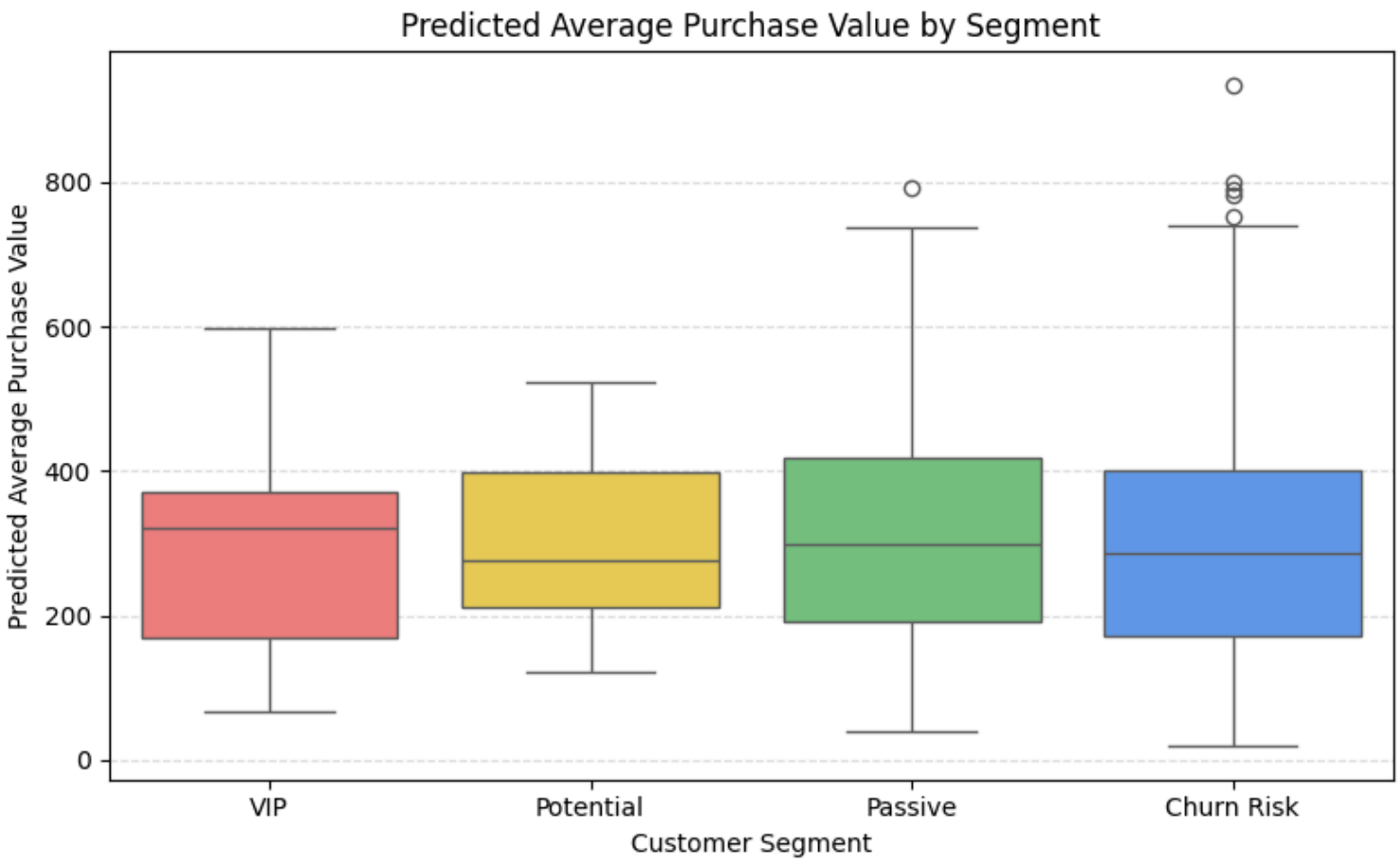


결과:

- 고객의 대부분은 평균 100~400달러 사이에서 구매할것으로 예측됨.
- 오른쪽으로 꼬리를 가진 분포 → 소수의 고객이 높은 금액(600~800+)을 지출할 가능성 있음.
- 전형적인 long-tail 분포로, 상위 고객 소수가 전체 수익을 가지는 영향력이 크다고 보임 → 해당 그룹 집중 육성 전략 필요

In [ ]:
```
import pandas as pd

temp = pd.DataFrame({
    'segment': data['segment'],
    'expected_avg_value': result,
```

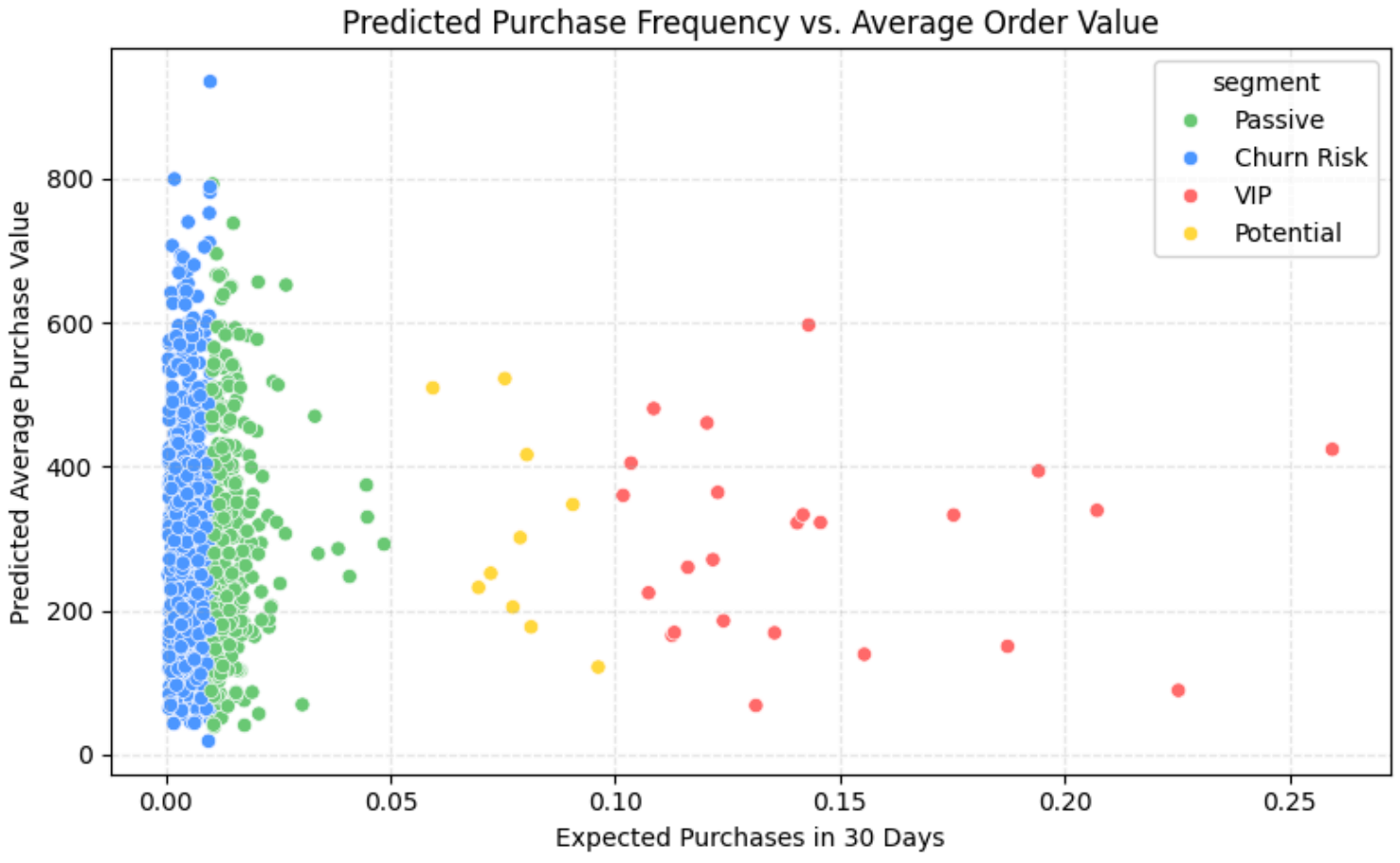
```
    'predicted_purchases_30d': data['predicted_purchases_30d']
})



plt.figure(figsize=(8, 5))
sns.boxplot(
    data=temp,
    x='segment',
    y='expected_avg_value',
    order=['VIP', 'Potential', 'Passive', 'Churn Risk'],
    palette=segment_palette
)

plt.title('Predicted Average Purchase Value by Segment')
plt.xlabel('Customer Segment')
plt.ylabel('Predicted Average Purchase Value')
plt.grid(axis='y', linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()
```



결과:

- 평균 구매 금액은 세그먼트 간 큰 차이는 없음.
- 오히려 Churn Risk 고객 중 일부가 높은 금액을 지출할 가능성이 있음 (상위 이상치).

In [ ]:
```
plt.figure(figsize=(8, 5))
sns.scatterplot(
    data=temp,
    x='predicted_purchases_30d',
    y='expected_avg_value',
    hue='segment',
    palette=segment_palette
)

plt.title('Predicted Purchase Frequency vs. Average Order Value')
plt.xlabel('Expected Purchases in 30 Days')
plt.ylabel('Predicted Average Purchase Value')
plt.grid(True, linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()
```



결과:

- 대부분의 고객은 왼쪽 하단 (재구매도가 적고, 금액도 낮음).
- VIP 고객군(빨강)은 예측 빈도와 금액이 모두 높은 이상적인 타겟.
- Potential 고객은 빈도는 낮지만 금액은 중간 이상 → 재활성화 가치 있음
- Passive/Churn Risk 고객 중에도 고금액 예상 고객 일부 존재 → 소극적 프리미엄 고객

구매 빈도와 구매 금액은 완전히 독립적이다 → 단순 세그먼트로만 마케팅 타겟을 정하면 기회를 놓칠 수 있음 → CLV 기반 이중 분석을 통해 고가치 군을 재발견할 수 있음

In [ ]:
```
# 전체 고객에 대한 예측 구매금액액 평균
# Expected conditional avg profit(모델이 예측한 평균금액) VS. Average profit(실제 데이터에서 측정된 평균금액)
print("Expected conditional average profit: %s, Average profit: %s" % (
    ggf.conditional_expected_average_profit(
        df['frequency'],
        df1['monetary_value']
    ).mean(),
    df[df['frequency']>0]['monetary_value'].mean()
))
```

```
Expected conditional average profit: 192.68685396004423, Average profit: 254.40209095919482
```

In [ ]: