

시니어 세대의 든든한 조력자

# 실버세이브

콜센터를 통해 통신 요금 할인부터 IPTV까지

회귀 예측을 통한 시니어 세대 맞춤 AICC 통신 결합 상품



# CONTENTS

## STEP 1

- 분석 기획
- 데이터 준비
- 데이터 분석

## STEP 2

- 문제점
- 원인 분석

## STEP 3

- 인사이트

## STEP 4

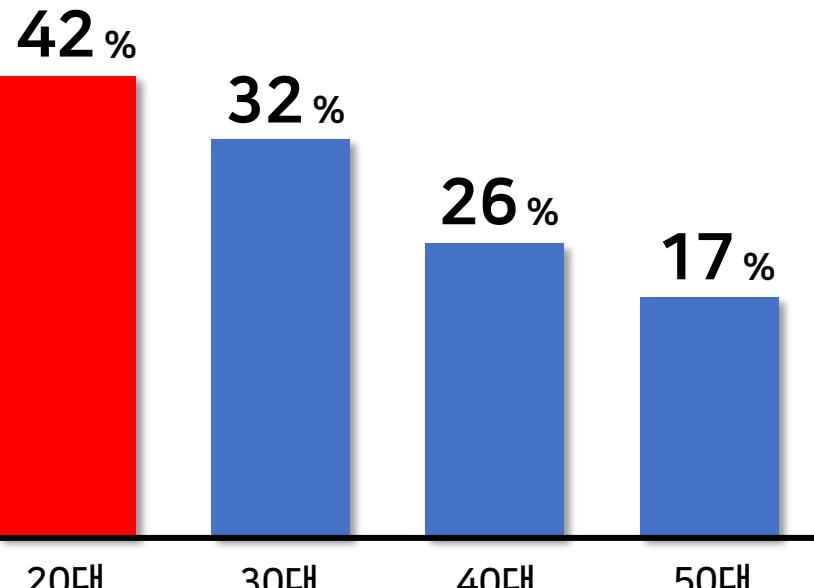
- 서비스 전략
- 기대효과

분석 기획

# 콜포비아, MZ세대 전화통화가 두렵다

## Fact Check

전화통화에 어려움을 느끼는 연령 비율



“전화보단 문자로 연락 주세요”  
MZ세대 10명 중 4명은 ‘콜 포비아’

MZ세대 2,735명 대상 설문 조사 결과, 전체의 42%는 콜 포비아 증상을 겪고 있다.

### 전화 통화시 어려움을 느끼는 이유

- 생각을 정리할 틈 없이 바로 대답해야 해서 59.1%
- 생각한 바를 제대로 말하지 못할 것이 걱정돼서 53.8%
- 문제, 메시지 등 텍스트로 소통하는 것이 편해서 46.6%
- 할 말이 떨어졌을 때 침묵이 불안해서 29.2%

데이터 준비

# 주요 Feature 파악

연령대	등록(계/피)	고용상태	성별	보험유지 기대확률	갱신	상품타입	교육수준	소득	...	결혼여부	월납입액	VOC	판매채널
1	계약자	고용	1	25.981	0	기본	대학졸업	864451 000	...	기혼	70000	0	인터넷
3	피보험자	고용	0	3.4410	1	중급	대학졸업	259620 00	...	미혼	80000	0	자사영업
3	피보험자	무직	1	9.0576	0	중급	대학졸업	0	...	미혼	90000	3	자사영업
4	계약자	고용	1	8.3472	0	중급	대학졸업	263660 00	...	기혼	100000	0	콜센터

데이터 준비

# 결측치 확인

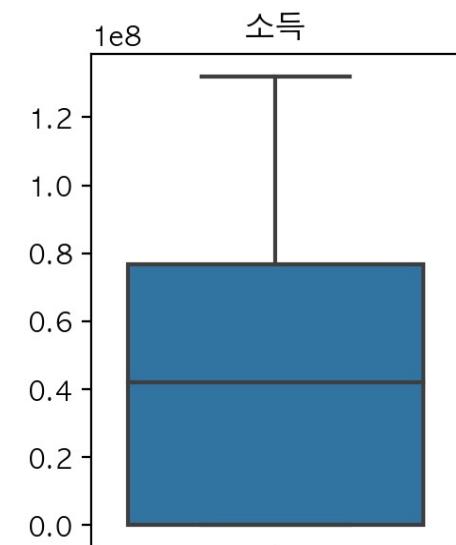
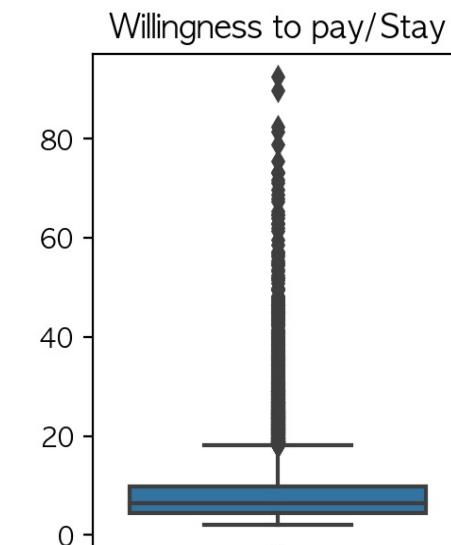
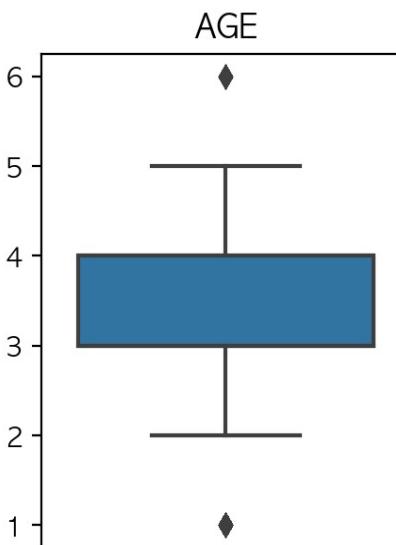
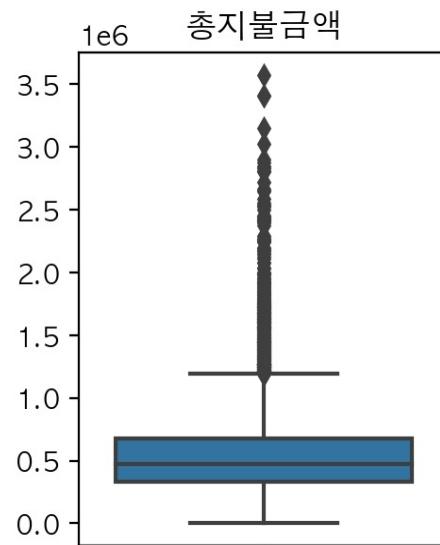
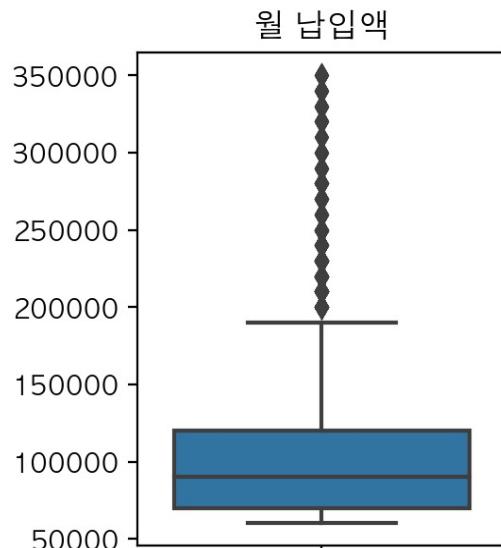


결측치 확인

```
1 missing_values = df.isnull().sum()  
2 columns_with_missing_values = missing_values[missing_values > 0]  
3  
4 print(columns_with_missing_values)  
5 print("결측치 없습니다.")
```

데이터 준비

# 수치형 변수 리스트



데이터 준비

# 이상치 확인



IQR(Interquartile Range) 방법을 사용하여 이상치 개수를 계산 및 각각의 수치형 컬럼에서 이상치를 count

```
1 def count_outliers(data, column_name):
2     Q1 = data[column_name].quantile(0.25)
3     Q3 = data[column_name].quantile(0.75)
4     IQR = Q3 - Q1
5
6     lower_bound = Q1 - 1.5 * IQR
7     upper_bound = Q3 + 1.5 * IQR
8
9     # lower bound와 upper bound 사이에 있는 값은 이상치가 아님
10    outliers_count = ((data[column_name] < lower_bound)
11                           | (data[column_name] > upper_bound)).sum()
12
13    return outliers_count
14
15 # 각각의 수치형 컬럼에서 이상치를 count
16 outliers_counts = {col: count_outliers(df, col) for col in numerical_columns}
17 outliers_counts
```

데이터 준비

# 컬럼 수정



## 컬럼 수정

```
1 # AGE 컬럼 수정: "1: 20대, 2: 30대, 3:40대, 4:50대, ..."를 실제 연령대로 변경  
2 df['AGE'] = (df['AGE'] * 10) + 10  
3  
4 # '타 상품 보유 현황' 컬럼에서 4이상의 값을 4로 변경  
5 df['타 상품 보유 현황'] = df['타 상품 보유 현황'].replace('4이상', '4')  
6 df['타 상품 보유 현황'] = df['타 상품 보유 현황'].astype(int)  
7 df['타 상품 보유 현황'] = df['타 상품 보유 현황'].apply(lambda x: min(x, 4))  
8  
9 df.head()
```

데이터 준비

# 데이터 구조 파악



데이터의 전반적 구조

```

1 data_info = pd.DataFrame()
2
3 data_info['Data Type'] = df.dtypes
4 data_info['Missing Values'] = df.isnull().sum()
5 data_info['Unique Values'] = df.nunique()
6
7 data_info

```

		Data Type	Missing Values	Unique Values
	clust	int64	0	6
	AGE	int64	0	6
	등록(계/피)	object	0	2
	고용상태	object	0	4
	성별	int64	0	2
	Willingness to pay/Stay	float64	0	3024
	갱신	int64	0	2
	상품타입	object	0	3
	교육수준	object	0	4
	소득	int64	0	7641
	지역	object	0	3
	결혼여부	object	0	2
	월 납입액	int64	0	30
	VOC	int64	0	6
	타 상품 보유 현황	int64	0	4
	온라인방문빈도	object	0	3
	갱신인센티브	object	0	4
	판매채널	object	0	4
	총지불금액	int64	0	1365
	자동차	object	0	5
	거주지사이즈	object	0	3

데이터 분석

# 상관관계 분석



상관관계 분석

```
1 # 자료형이 object인 컬럼을 찾고, 수치형으로 변환할 수 있는지 확인
2 df_corr = df.copy()
3
4 object_columns = df_corr.select_dtypes(include=['object']).columns
5
6 # object 컬럼에서 unique value를 확인하고, 수치형으로 변환할 수 있는지 확인
7 object_column_values = {}
8 for col in object_columns:
9     try:
10         df_corr[col] = pd.to_numeric(df_corr[col], errors='raise')
11     except Exception:
12         object_column_values[col] = df_corr[col].unique()
13
14 object_column_values
```

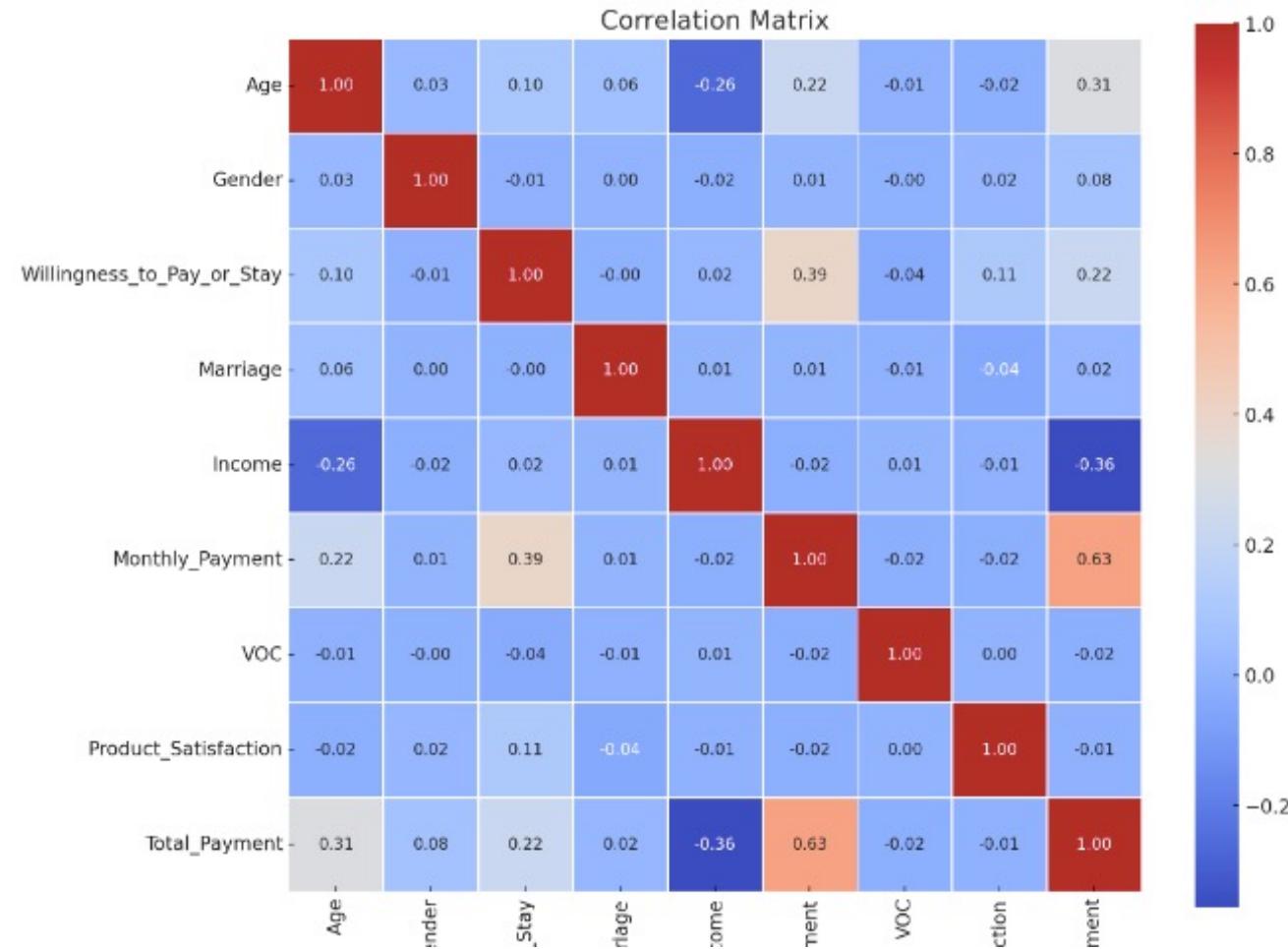
데이터 분석

# 상관관계 분석

```
{'등록(계/피)': array(['피보험자', '계약자'], dtype=object),  
'고용상태': array(['무직', '고용', '휴직', 'FALSE'], dtype=object),  
'상품타입': array(['기본', '고급', '중급'], dtype=object),  
'교육수준': array(['대학졸업', '고졸이하', '석사', '박사'], dtype=object),  
'지역': array(['도시근교', '시골', '도심'], dtype=object),  
'결혼여부': array(['미혼', '기혼'], dtype=object),  
'온라인방문빈도': array(['자주방문', '비방문', '종종'], dtype=object),  
'갱신인센티브': array(['없음', '포인트', '할인', '설계사독려'], dtype=object),  
'판매채널': array(['자사영업', '인터넷', '대리점', '콜센터'], dtype=object),  
'자동차': array(['일반세단', '컴팩트카', 'SUV', '스포츠카', '고급차'], dtype=object),  
'거주지사이즈': array(['소', '중', '대'], dtype=object)}
```

데이터 분석

# 상관관계 분석



데이터 분석

# 상관관계 분석



상관관계 분석

```
1 # 변환할 열 및 해당 변환 함수 정의
2 conversion_columns = {
3     'Willingness to pay/Stay': lambda x: np.nan if x is np.nan else
4         (float(x.strip('%')) / 100 if isinstance(x, str) else x),
5     '소득': lambda x: np.nan if x is np.nan else (int(x.replace(',', '')) if
6         isinstance(x, str) else x),
7     '월 납입액': lambda x: np.nan if x is np.nan else (int(x.replace(',', '')) if
8         isinstance(x, str) else x),
9     '총지불금액': lambda x: np.nan if x is np.nan else (int(x.replace(',', '')) if
10        isinstance(x, str) else x)
11 }
12
13 # 변환 함수를 해당 열에 적용, 컬럼 개체 유형 확인
14 for col, func in conversion_columns.items():
15     if df_corr[col].dtype == 'object':
16         df_corr[col] = df_corr[col].map(func)
17
18 df_corr
```

## 데이터 분석

# 상관관계 분석

clust	AGE	등록(계/피)	고용상태	성별	Willingness to pay/Stay	갱신	상품타입	교육수준	소득	...	결혼여부	월 납입액	voc	타 상품	보유	현황	온라인방문빈도	갱신인센티브	판매채널	총지불금액	자동차	거주지사이즈
0	2	40	피보험자	무직	1	2.616381	0	기본 대학졸업	0	...	미혼	80000	0		1	자주방문	없음	자사영업	631000	일반세단	소	
1	4	20	계약자	고용	0	6.352530	0	기본 고졸이하	102887400	...	미혼	80000	1		4	비방문	포인트	인터넷	54000	일반세단	중	
2	5	30	계약자	휴직	0	4.974354	0	기본 대학졸업	22159500	...	기혼	60000	0		4	비방문	할인	자사영업	362000	일반세단	중	
3	0	60	계약자	고용	1	13.480284	0	고급 고졸이하	51562500	...	기혼	110000	0		2	비방문	할인	자사영업	1264000	컴팩트카	대	
4	3	50	계약자	고용	0	9.776436	0	기본 석사	26820200	...	기혼	120000	0		3	비방문	없음	대리점	947000	SUV	중	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
11995	0	20	계약자	고용	1	25.985100	0	기본 대학졸업	86451000	...	기혼	70000	0		2	비방문	설계사독려	인터넷	245000	일반세단	중	
11996	0	40	피보험자	고용	0	3.441000	1	중급 대학졸업	25962000	...	미혼	80000	0		1	자주방문	할인	자사영업	468000	일반세단	중	
11997	2	40	피보험자	무직	1	9.057600	0	중급 대학졸업	0	...	미혼	90000	3		2	자주방문	할인	자사영업	977000	일반세단	중	
11998	4	50	계약자	고용	1	8.347200	0	중급 대학졸업	26366000	...	기혼	100000	0		3	비방문	포인트	자사영업	854000	일반세단	대	
11999	2	40	피보험자	무직	1	2.897100	0	중급 대학졸업	0	...	미혼	80000	0		1	자주방문	없음	콜센터	456000	컴팩트카	중	

12000 rows × 21 columns

데이터 분석

# 상관관계 분석



상관관계 분석

```
1 # 남아있는 object 타입의 열들을 원-핫 인코딩
2 data_encoded = pd.get_dummies(df_corr, columns=[col for col in object_columns if col
not in conversion_columns])
3
4 # 변환된 데이터 확인
5 data_encoded.head()
6
7 # 상관 관계 행렬 계산
8 correlation_matrix = data_encoded.corr()
9
10 # "Willingness to pay/Stay" 열과의 상관 관계 값을 추출
11 wtp_correlation = correlation_matrix['Willingness to pay/Stay'].drop('Willingness to
pay/Stay')
12
13 # "Willingness to pay/Stay"와 절대 상관 관계가 가장 높은 열 찾기
14 max_corr_column = wtp_correlation.abs().idxmax()
15 max_corr_value = wtp_correlation[max_corr_column]
16
17 max_corr_column, max_corr_value
```

데이터 분석

# 상관관계 분석



## 상관관계 분석

```
1 # "월 납입액" 열과 다른 모든 열 간의 상관 관계 값을 추출  
2 monthly_payment_correlation = correlation_matrix['월 납입액'].drop('월 납입액')  
3  
4 # "월 납입액"과 절대 상관 관계가 가장 높은 열 찾기  
5 max_corr_column_payment = monthly_payment_correlation.abs().idxmax()  
6 max_corr_value_payment =  
7 monthly_payment_correlation[max_corr_column_payment]  
8 max_corr_column_payment, max_corr_value_payment
```

('월 납입액', 0.39182727428367115)  
(‘자동차\_고급차’, 0.6867684497216702)

데이터 분석

# 상관관계 분석



상관관계 분석

```

1 # 변환된 "판매채널" 열들을 추출
2 sales_channel_columns = [col for col in data_encoded.columns if '판매채널' in col]
3
4 # 각 "판매채널" 열과 절대 상관 관계가 가장 높은 열 찾기
5 max_corr_sales_channel = {}
6 for col in sales_channel_columns:
7     column_correlation = correlation_matrix[col].drop(col)
8     max_corr_column = column_correlation.abs().idxmax()
9     max_corr_value = column_correlation[max_corr_column]
10    max_corr_sales_channel[col] = {'column': max_corr_column, 'value':
11 max_corr_value}
12 max_corr_sales_channel

```

```

{'판매채널_대리점': {'column': '판매채널_자사영업', 'value': -0.48986326254258433},
'판매채널_인터넷': {'column': '판매채널_대리점', 'value': -0.3242286564158329},
'판매채널_자사영업': {'column': '판매채널_대리점', 'value': -0.48986326254258433},
'판매채널_콜센터': {'column': '판매채널_대리점', 'value': -0.38533700462276743}}

```

데이터 분석

## 상관관계 분석

- Willingness to pay/Stay와 가장 상관관계가 높은 컬럼은 월 납입액이다.(0.39)  
→ 월 납입액이 높을 수록 계약 유지율이 높다.
- 월 납입액과 가장 상관관계가 높은 컬럼은 자동차\_고급차이다.(0.69)  
→ 고급차를 소유하고 있을수록 월 납입액이 많다.
- 판매채널에서 대리점이 다른 요소보다 음의 상관관계가 높다.

데이터 분석

# 다중 공선성 파악



다중 공선성 파악

```
1 object_columns = df_corr.select_dtypes(include=['object']).columns
2 df_corr = pd.get_dummies(df_corr, columns=object_columns, drop_first=True)
3
4 numeric_data = df_corr.select_dtypes(exclude=["object"])
5 vif_data = pd.DataFrame()
6 vif_data["VIF Factor"] = [variance_inflation_factor(numeric_data.values, i)
7                           for i in range(numeric_data.shape[1])]
8 vif_data["features"] = numeric_data.columns
9 vif_data = vif_data.sort_values("VIF Factor",
10 ascending=False).reset_index(drop=True)
11 vif_data
```

## 데이터 분석

# 다중 공선성 파악

	VIF Factor	features		VIF Factor	features
0	inf	온라인방문빈도_자주방문	18	2.936406	Willingness to pay/Stay
1	inf	등록(계/피)_피보험자	19	2.921016	거주지사이즈_소
2	48.003530	월 납입액	20	2.864699	자동차_고급차
3	31.492159	고용상태_고용	21	2.566097	갱신인센티브_할인
4	25.485837	AGE	22	2.007325	결혼여부_미혼
5	13.890133	총지불금액	23	1.985625	성별
6	11.717200	소득	24	1.813598	지역_도심
7	11.273184	고용상태_무직	25	1.748418	판매채널_자사영업
8	9.786963	상품타입_기본	26	1.564705	갱신인센티브_포인트
9	7.813972	타 상품 보유 현황	27	1.545466	판매채널_콜센터
10	7.753694	거주지사이즈_중	28	1.418189	판매채널_인터넷
11	6.211034	자동차_일반세단	29	1.407101	갱신인센티브_없음
12	5.719396	clust	30	1.389449	갱신
13	4.263100	상품타입_중급	31	1.369704	교육수준_석사
14	4.049870	고용상태_휴직	32	1.321694	자동차_스포츠카
15	3.184604	지역_시골	33	1.190587	VOC
16	3.116798	교육수준_대학졸업	34	1.161174	교육수준_박사
17	3.105083	자동차_컴팩트카	35	1.058547	온라인방문빈도_종종

데이터 분석

# 판매채널별 데이터 수

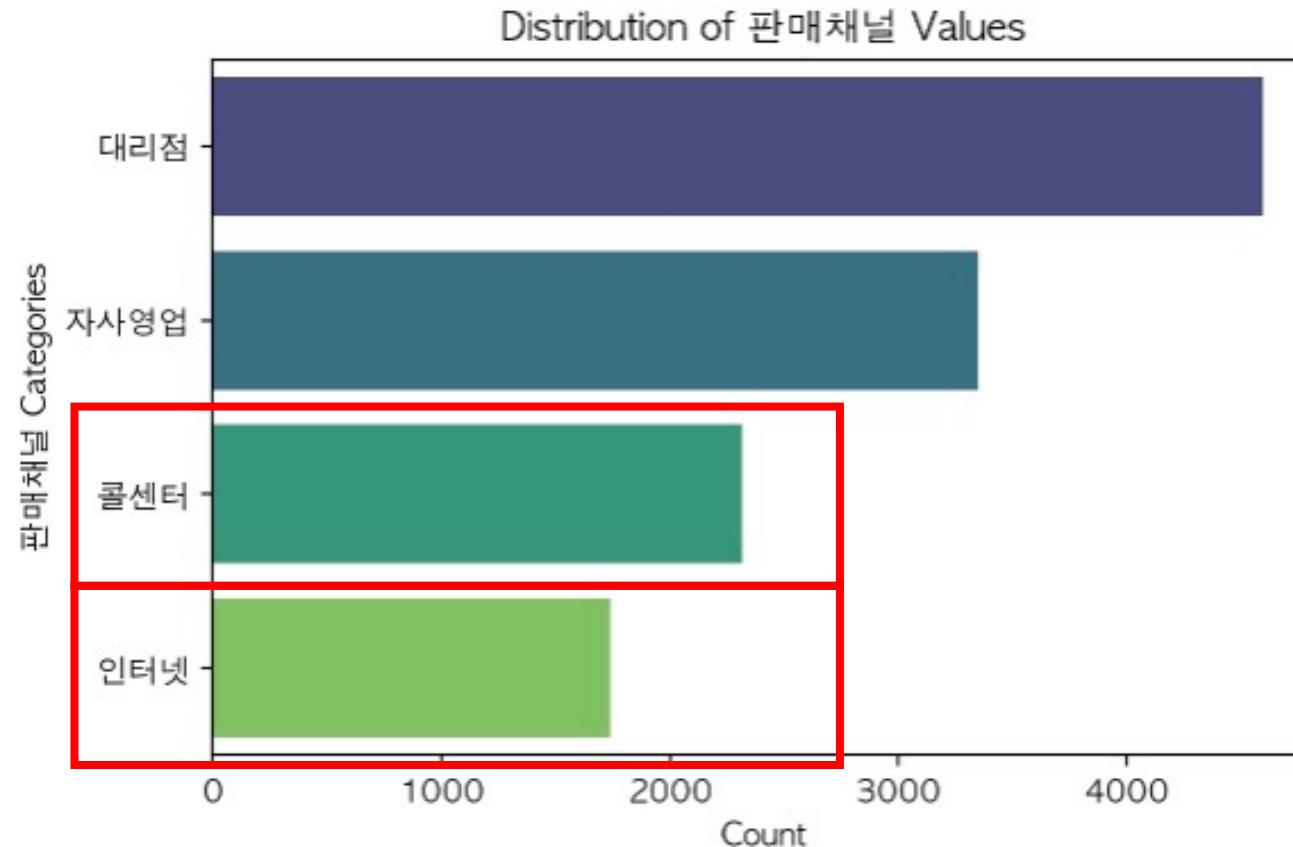


판매채널별 데이터 수

```
1 sales_channel = df['판매채널']
2
3 plt.figure(figsize=(6, 4))
4
5 sns.countplot(y=sales_channel, order=sales_channel.value_counts().index,
6                 palette="viridis")
6 plt.title('판매채널 값 분포')
7 plt.xlabel('Count')
8 plt.ylabel('Categories')
9 plt.tight_layout()
10 plt.show()
```

데이터 분석

# 판매채널별 데이터 수



데이터 분석

# 데이터셋 평균값 계산



데이터셋 평균값 계산

```
1 # 전체 데이터셋의 평균값 계산: data_encoded의 모든 컬럼에 대한 평균값을 계산
2 overall_mean = data_encoded.mean()
3
4 # "콜센터"와 "인터넷" 부분집합의 평균값 계산
5 # "판매채널"이 "콜센터" 또는 "인터넷"인 데이터의 부분집합에 대한 각 컬럼의 평균값을 계산
6 call_center_mean = data_encoded[data_encoded['판매채널_콜센터'] == 1].mean()
7 internet_mean = data_encoded[data_encoded['판매채널_인터넷'] == 1].mean()
8
9 # 평균값 비교를 위한 DataFrame 생성
10 # 전체 데이터셋의 평균값과 "콜센터", "인터넷" 부분집합의 평균값을 비교하기 위한 DataFrame을 생성
11 df_comp = pd.DataFrame({'Overall': overall_mean,
                           'Call Center': call_center_mean,
                           'Internet': internet_mean})
12
13
14
15 # 부분집합과 전체 데이터셋 간의 평균값 차이 계산
16 # "콜센터"와 "인터넷" 부분집합의 각 컬럼의 평균값과 전체 데이터셋의 평균값과의 차이를 계산
17 df_comp['Call Center Difference'] = df_comp['Call Center'] -
18 df_comp['인터넷Difference'] = df_comp['Internet'] - df_comp['Overall']
```

데이터 분석

# 데이터셋 평균값 계산

판매채널_대리점	-0.382667	-0.382667
판매채널_인터넷	-0.145000	0.855000
판매채널_자사영업	-0.279083	-0.279083
판매채널_콜센터	0.806750	-0.193250

데이터 분석

# 콜센터와 인터넷 채널을 통한 고객 특성 파악

- Willingness to pay/Stay
  - 콜센터를 통한 고객들은 전체 평균보다 약 14.4% 높은 결제 의향이 있으며, 인터넷을 통한 고객들은 약 33.7% 낮은 결제 의향을 보인다.
  - 콜센터를 통해 연락하는 고객들이 상품에 대한 더 높은 흥미나 만족도를 가질 수 있다.
- 소득
  - 콜센터를 통한 고객들의 평균 소득은 전체 평균보다 약 834,299원 높으며, 인터넷을 통한 고객들의 평균 소득은 약 785,591원 높다.
- 월 납입액
  - 콜센터를 통한 고객들의 평균 월 납입액은 전체 평균보다 약 449.6원 낮으며, 인터넷을 통한 고객들의 평균 월 납입액은 약 916.4원 낮다.
- 판매채널\_콜센터
  - 콜센터를 통한 고객들은 전체 평균보다 약 80.7% 높은 비율을 차지하며, "인터넷"을 통한 고객들은 이 비율이 약 19.3% 낮다.
  - 이는 콜센터를 통해 연락하는 고객들이 데이터셋에서 상대적으로 더 많다는 것을 의미한다.

데이터 분석

# 판매채널별 갱신 수

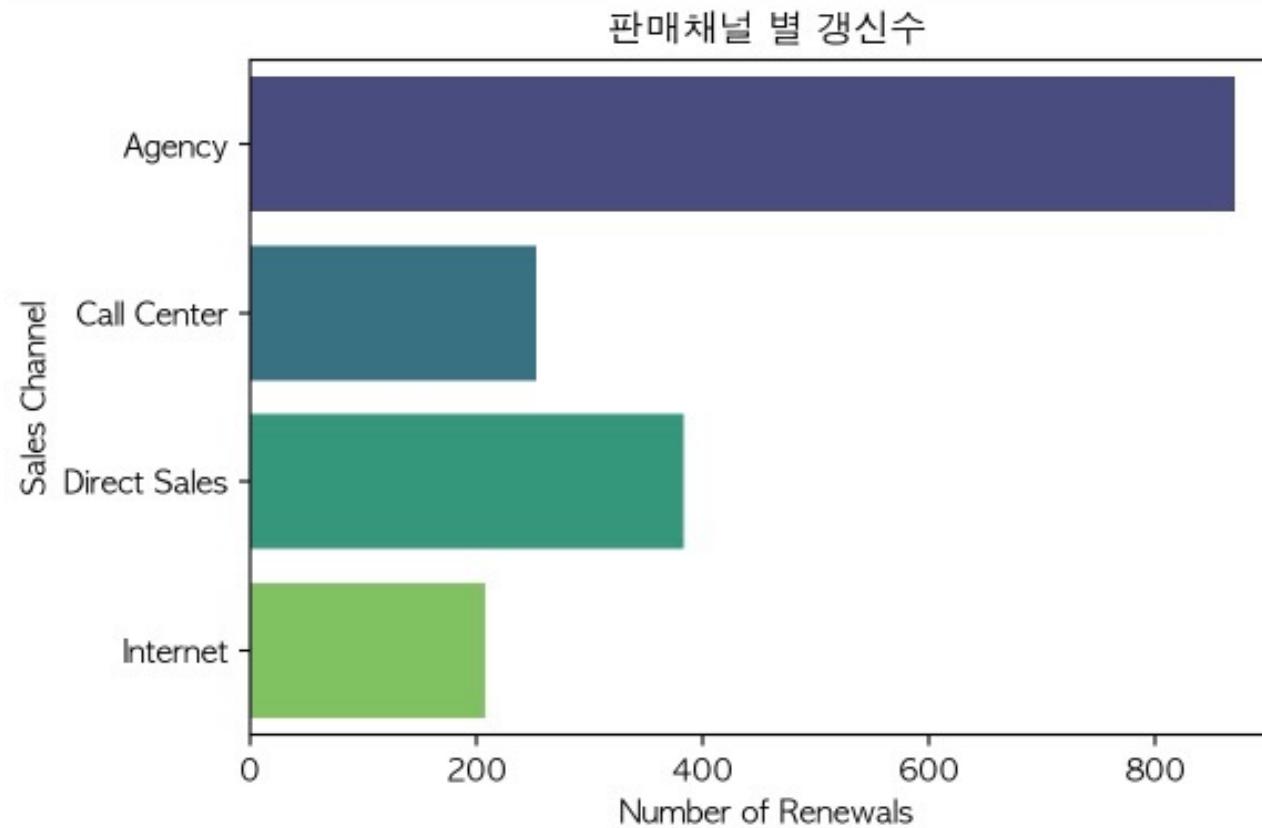


판매채널별 갱신수

```
1 renewal_data = df[['판매채널', '갱신']].copy()
2 renewal_data['판매채널'] = renewal_data['판매채널'].map(sales_channel_mapping)
3 renewal_counts = renewal_data.groupby('판매채널')['갱신'].sum().reset_index()
4
5 plt.figure(figsize=(6, 4))
6 sns.barplot(x='갱신', y='판매채널', data=renewal_counts, palette='viridis')
7 plt.title('판매채널 별 갱신수')
8 plt.xlabel('Number of Renewals')
9 plt.ylabel('Sales Channel')
10 plt.tight_layout()
```

데이터 분석

# 판매채널별 갱신 수



데이터 분석

# 판매채널별 갱신 수에 따른 보험유지 기대확률

LinearRegression

```

● ● ●
LinearRegression

1 from sklearn.model_selection import train_test_split
2 from sklearn.linear_model import LinearRegression
3 from sklearn.metrics import mean_squared_error, r2_score
4
5 # 특성과 목표 변수 추출
6 X = data_encoded.drop(columns=['Willingness to pay/Stay'])
7 y = data_encoded['Willingness to pay/Stay']
8 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)
9
10 model = LinearRegression()
11 model.fit(X_train, y_train)
12
13 y_pred = model.predict(X_test)
14
15 # R2 스코어와 평균 제곱 오차(MSE)를 계산하여 모델 성능 평가
16 r2 = r2_score(y_test, y_pred)
17 mse = mean_squared_error(y_test, y_pred)

```

r2\_score: 0.21

→ 그다지 성능이 좋지 못함

mean\_squared\_error: 47

→ 실제 값과 예측 값 사이의 차이가 큼

데이터 분석

# 판매채널별 갱신 수에 따른 보험유지 기대확률

RandomForest



RandomForest

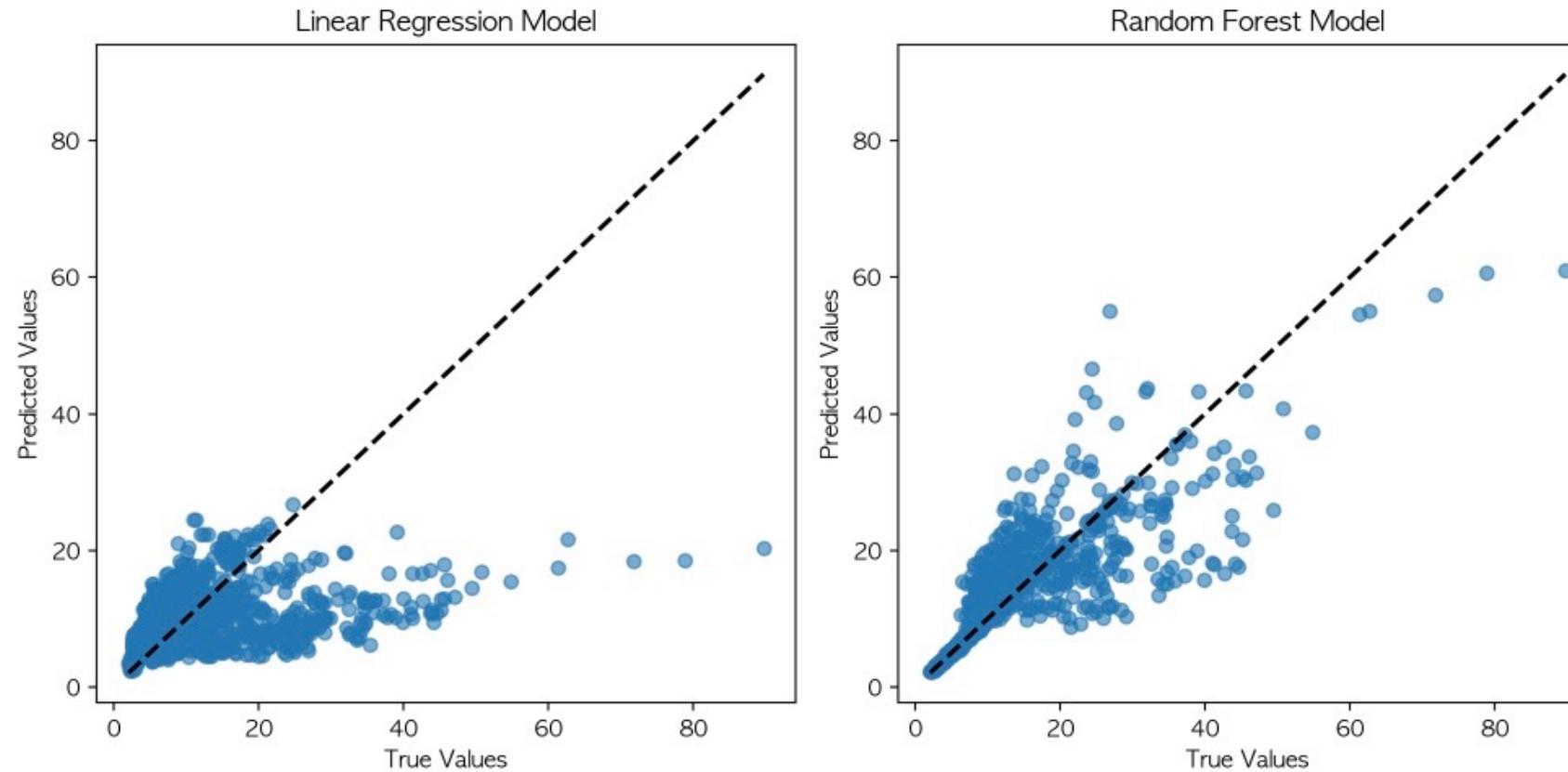
```
1 from sklearn.ensemble import RandomForestRegressor
2 from sklearn.preprocessing import StandardScaler
3 from sklearn.pipeline import Pipeline
4
5 scaler = StandardScaler()
6
7 rf_model = RandomForestRegressor(random_state=42, n_jobs=-1)
8
9 pipeline = Pipeline([('scaler', scaler), ('model', rf_model)])
10 pipeline.fit(X_train, y_train)
11
12 y_pred_rf = pipeline.predict(X_test)
13
14 r2_rf = r2_score(y_test, y_pred_rf)
15 mse_rf = mean_squared_error(y_test, y_pred_rf)
```

r2\_score: 0.77  
→ 성능 좋음

mean\_squared\_error: 13.6  
→ 쓸만함

데이터 분석

# 판매채널별 갱신 수에 따른 보험유지 기대확률



문제점

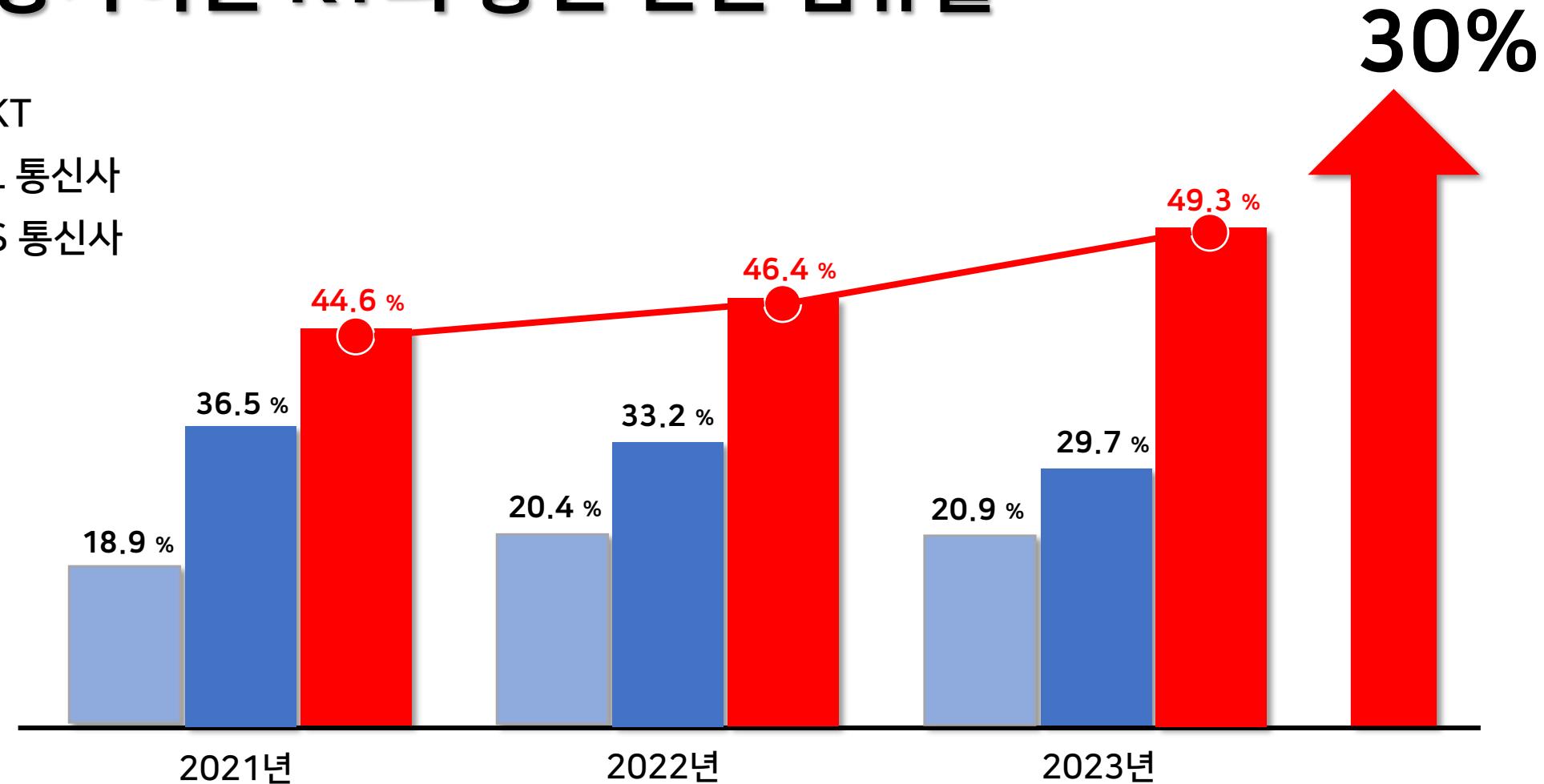
## 인터넷에 익숙하지만 전화는 두려운 MZ세대

인터넷에 익숙한 MZ세대는 굳이 전화를 하지 않는다.  
좋은 상품을 알아서 찾는다.

원인 분석

# 매년 증가하는 KT의 통신 민원 점유율

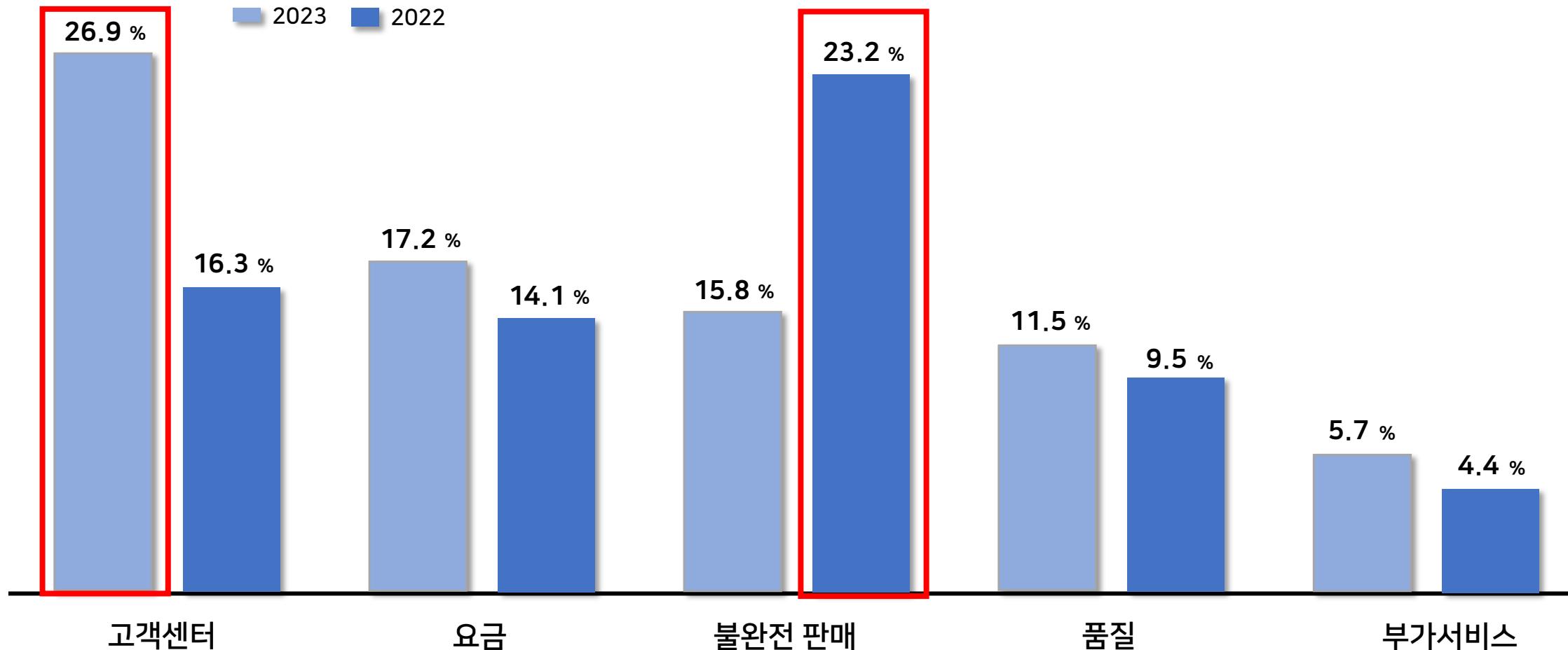
- KT
- L 통신사
- S 통신사



출처: 소비자민원평가

원인 분석

# 통신사 민원 접수 유형



출처: 소비자민원평가

원인 분석

## 고령층 고객센터, 불완전판매 피해 크게 증가

최근 3년간 만 65세 이상 고령소비자 이동전화서비스 관련 피해구제 신청 횟수 총 **437**건

가입단계 피해 287건, 이용단계 105건으로 전체 비율의 **90%** 차지



김 모씨

8개월 사용한 휴대전화가 잘 안돼 문의하려고 KT 대리점을 방문했다가 모친도 모르는 사이에 새로운 기기로 개통됐다

**계약서 글자도 잘 안보이는 82세 할머니에게 사기나  
다름 없는 식으로 개통을 시켰다**며 해결을 촉구했다



홍 모씨

LGU+고객센터로부터 기존 이용 중인 '가족결합'보다 새로 나온 상품이 요금 할인을 더 받을 수 있다는 안내를 받고 가족 모두 변경했다. 홍 씨의 경우 7만 원 정도의 요금이 나올 것으로 안내 받았으나 고지서를 받아보니 12만 원이 청구됐다.

**"고객센터에서 잘못 안내해 요금을 더 내게 됐는데 원상  
복구도 어렵다"**며 불만을 토로했다



고령층 ㄱ씨

지난해 6월 한 통신사의 대리점 직원으로부터 '월 6만원 대의 요금제를 4개월 유지한 후 낮은 요금제로 변경해준다'는 안내를 받고 이동전화 이용 계약을 체결했다.

하지만 8만원대의 요금이 청구돼 해지를 요구했지만 해당 통신사 대리점은 ㄱ씨에게 **'제휴카드 혜택할인을 적용한 실부담 금액을 설명한 것이며 고객확인안내서에 최  
모씨의 자필 서명이 확인된다'**는 사유로 이를 거부했다.



고령층 ㄴ씨

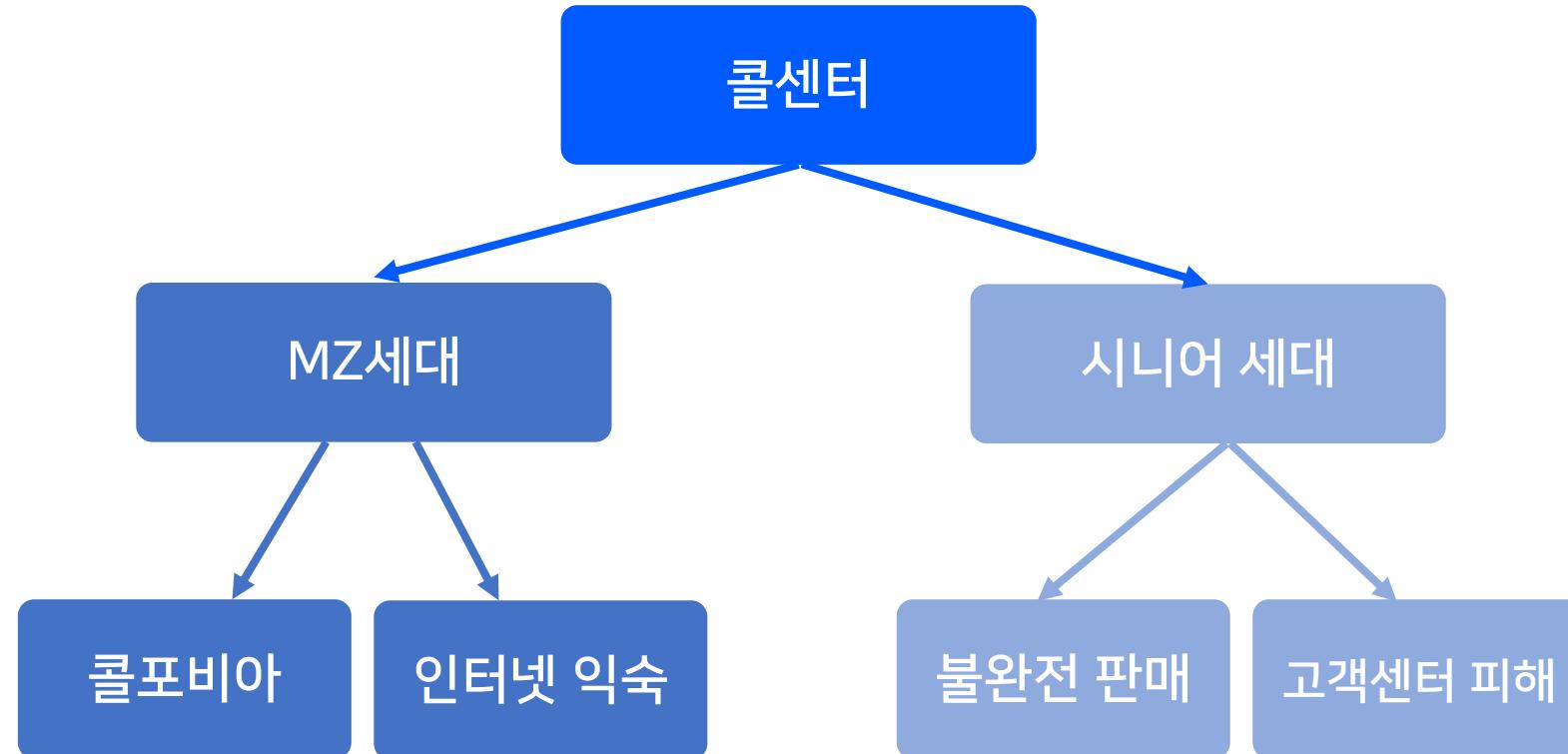
한 통신사 대리점 직원으로부터 '기존 휴대폰은 카카오톡이 불가하기 때문에 휴대폰을 교체하라'는 안내를 받고 신규 휴대폰 구입과 이동전화서비스 계약을 체결했다.

하지만 **기존 단말기에서도 카카오톡이 된다는 사실을 확인한 후 해당 통신사 대리점에 개통철회를 요구했지만 이미 휴대폰을 사용했다는 이유로 거부됐다.**

인사이트

# KT, 고객의 고객을 위한 고객중심 기업

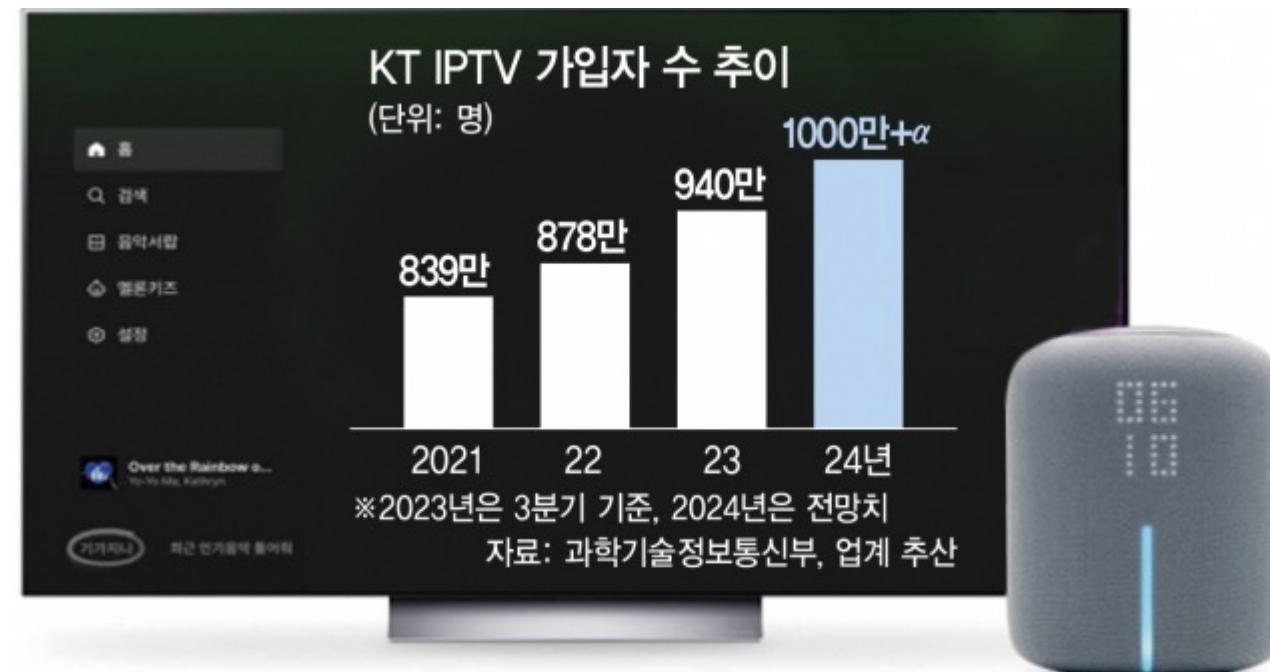
콜센터 프로모션을 통한 상품의 완전 판매



인사이트

# KT만의 확실한 필살기

국내 1위 최대/최다 IPTV 가입자 수 확보 기업



인사이트

# 연령별 TV/스마트폰 이용빈도

방통위, 방송매체 이용행태 조사

TV(주5일 이상 이용)

스마트폰(주5일 이상 이용)

98.6%

70대 이상

N=848

96.6%

60대  
N=1034

90.0%

50대  
N=1241

79.8%

40대  
N=1168

67.8%

30대  
N=967

41.4%

20대  
N=969

25.2%

10대  
N=480

50.7%

87.7%

95.1%

96.6%

99.4%

98.8%

99.2%

A photograph of a happy senior couple sitting on a light-colored sofa. The man, on the left, is wearing a brown sweater and has his arm around the woman. The woman, on the right, is wearing a pink cardigan over a white top and is holding a black smartphone. They are both smiling and looking at the screen of the phone. In the background, there are large windows showing greenery outside.

실버 요금제 + IPTV  
AICC 통신 결합 상품

# 실버세이브

서비스 전략

**실버 요금제 + IPTV**

**AICC 통신 결합 완전 상품**

# 실버세이브

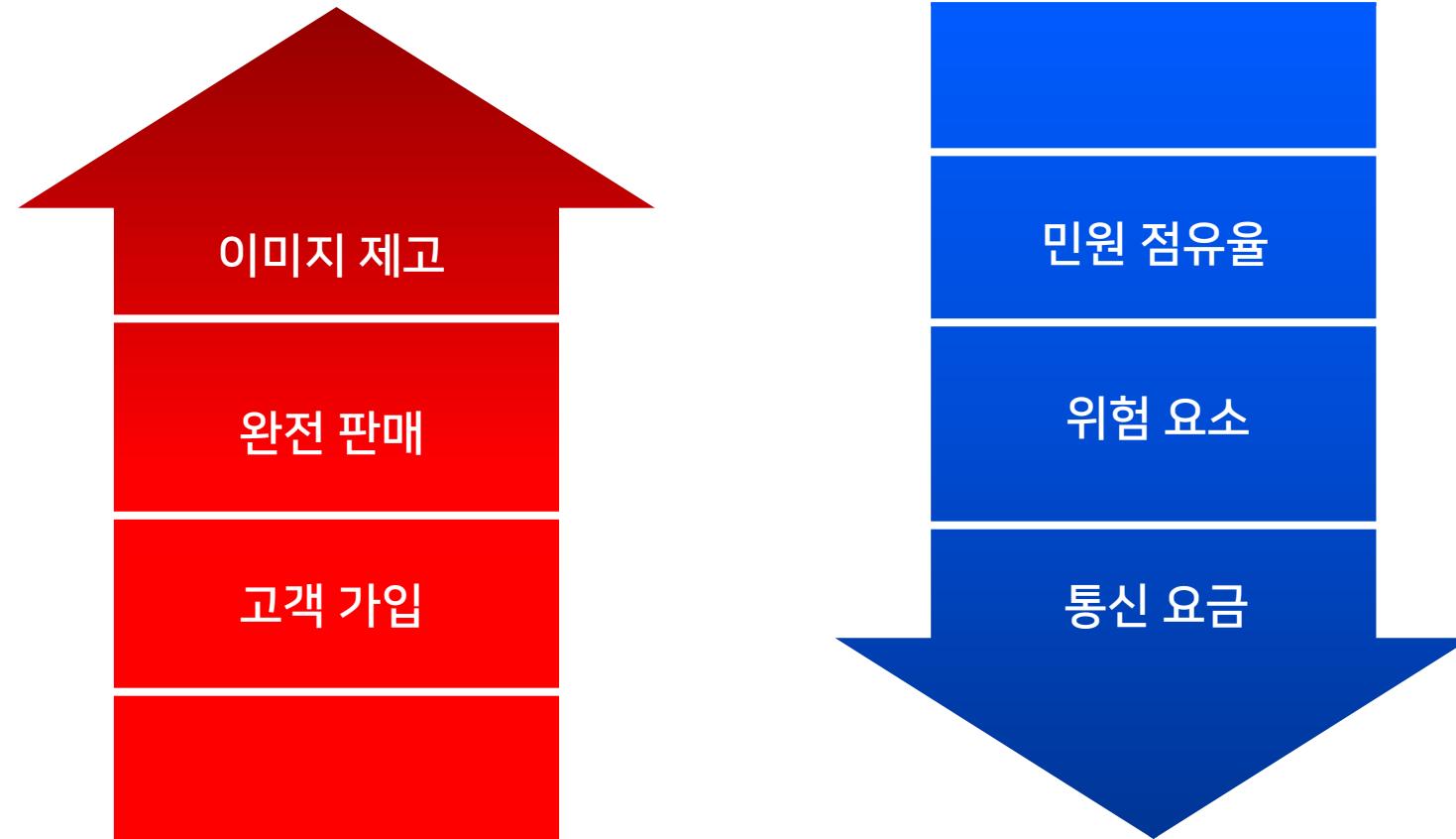
AICC를 통해 직접 통화 후 상품 가입

**전화통화로 가입은 쉽게,  
통신비 할인과 혜택으로 채우다**



기대효과

# 요금은 확 줄이고, 혜택은 확 늘리고!



시니어의 든든한 조력자

**kt** 실버세이브가 함께 하겠습니다