로지스틱 회귀분석을 이용한
자동차보험 가입 여부 예측

회귀분석II 학기말 과제

2018580004 통계학과 김다희

목차

1. 서론

1.1 연구목적

본 연구의 목적은 크게 두 가지로, 첫 번째는 한 건강보험회사 고객의 데이터를 이용해 기존 고객이 해당 회사에서 앞으로 제공할 **자동차보험에 가입할지에 대한 예측 모형**을 로지스틱 회귀분석을 이용해 적합해 보는 것이다. 고객이 어떤 상품에 관심을 가질지 예측하는 모델을 구축하는 것은 기업의 의사결정에 매우 중요한 역할을 한다. 이러한 예측을 통해 개별 고객의 특성에 맞는 마케팅 전략을 계획해 효율성 있는 비즈니스 모델을 세울 수 있으며, 비용 대비 수익을 최적화 할 수 있기 때문이다.

두 번째는 로지스틱 회귀모형을 이용한 예측에서 끝나지 않고, 각각의 회귀계수가 어떤 의미를 가지는지, 그 결과가 실생활에서의 상식과 일치하는지 등 **로지스틱 모형의 이해와 해석**을 하고자 한다. 해당 데이터를 가지고 최적의 예측 모델을 찾는 방법에는 로지스틱 모형 외에도 다양한 것들이 있지만, 이번 분석에서는 최적의 모델을 찾는 것보다는 로지스틱 모형으로 적합을 시도해보고 그 결과를 이해하는 데 의의를 두고자 한다.

즉, 다시 말해서 이번 연구의 목표는 건강보험 회사의 고객 정보를 바탕으로 자동차보험 가입에 대한 예측 모형을 로지스틱 회귀분석 방법으로 적합 후 이해하고 해석하는 것이다.

1.2 문헌 연구

이번 연구에서는 로지스틱 모형으로의 적합만을 시도하지만, 결국 이번 데이터 분석의 궁극적인 목표는 자동차보험 가입에 영향을 미치는 변수를 찾고, 이를 이용해 회사의 상품이 가장 효율적으로 많이 팔리도록 하는 데에 있다. 따라서 이런 보험의 판매 및 마케팅에 관련된 몇 가지 문헌을 찾아보았으며 문헌은 아니지만, 로지스틱 회귀모형 이외에 다른 분석법을 이용한 예측을 시도한 분석의 결과도 참고용으로 가져왔다.

1.2.1 자동차보험과 산업재해보상보험과의 관계에 대한 연구

이번 분석에서는 단순히 '자동차보험이 많이 팔릴 요인'을 찾는 것이 아니라, '건강보험에 가입한 고객 중 자동차보험에도 가입할 고객'을 찾는 것이므로 자동차보험과 관련 건강보험의 관계에 대한 연구를 참고 할 수 있을 것 같다.

다만 이 논문은 근본적인 산업재해보상보험과 자동차보험의 역할과 보상 범위 등 실무적인 차원에서의 접근이기 때문에 내용 자체에서 어떤 요인이 자동차보험의 판매에 영향을 미칠지를 바로 끌어내기는 어려워 보인다. 이는 말 그대로 자동차보험과 건강보험의 관계 정도의 해석에만 도움을 줄 수 있을 듯하다.

1.2.2 자동차보험의 마케팅 전략에 관한 연구

이 논문은 자동차보험의 마케팅에 관한 좀 더 포괄적인 내용을 통해 어떻게 하면 자동차보험을 더 많이, 더 효율적으로 팔 수 있는지 비즈니스 전략적 시점으로 분석한 연구이다. 주로 보험마케팅에 관한 고찰과 우리나라 자동차 시장의 실태에 대한 분석을 바탕으로 자동차보험의 마케팅 전략을 도출하고 있다.

이는 이번 분석과는 결이 조금 다르지만, 최종적인 분석 이유인 자동차보험을 '잘' 파는 데까지 활용되기 위해서는 참고할 수 있을 것 같다. 조금 아쉬운 점은 2001년도의 논문이기 때문에, 현 상황에 그대로 적용하기엔 괴리감이 있을 듯하다.

1.2.3 Vehicle Insurance EDA and boosting models

이는 데이터를 제공한 사이트(캐글)에 올라온 해당 데이터에 대한 다양한 분석 결과 중, 가장 많은 인기를 얻은 분석 방법을 가지고 온 것이다. 여기선 로지스틱 모형 대신에 랜덤 포레스트와 XGBoost 분류기 등의 부스팅 방법의 분류 모델을 이용해 예측 모형을 적합하고 있다. 대부분의 방법이 AUC=0.85~0.87 정도의 성능을 내는데, 직접 로지스틱 모형으로 적합 후 비교해 볼 수 있을 것 같다.

1.3 데이터 설명

1.3.1 데이터 소개

위에서 언급했듯이 이번 연구에서 분석할 데이터는 **건강보험을 제공하는 보험회사의 고객에 대한 데이터**로, 고객에 대한 개인정보를 바탕으로 보험회사가 새로 제공할 자동차보험에 관심을 가질지 예측하는 모델을 구축하기 위해 제공되었다. 이는 캐글에서 얻을 수 있고, 정확한 출처는 참고문헌에서 확인할 수 있다.

해당 회사에서는 train data와 test data를 모두 제공하고 있지만, 본 연구에서는 예측의 향상을 위한 분석이 아닌 '로지스틱 모형'의 적합을 시도하는 데 의의를 두고 있기 때문에 train data 내에서 데이터를 train data와 test data로 나눈 뒤 예측의 성능을 확인하는 과정까지만 다룰 예정이다.

따라서 본 연구에서는 제공된 세 가지 데이터(sample_submission.csv, test.csv, train.csv) 중 train.csv만을 이용할 것이다. 데이터 용량은 20.44MB로, 12개의 변수와 381,109개의 관측으로 이루어졌으며 결측값과 중복값은 없다. 자세한 건 분석과정에서 확인하도록 하고, 이어서 변수를 살펴보겠다.

1.3.2 변수 소개

변수	변수 설명
1. id	고객의 ID
2. Gender	고객의 성별(Male/Female)
3. Age	고객의 나이
4. Driving_License	운전면허 유무(0:없음/1:있음)
5. Region_Code	지역 코드
6. Previously_Insured	자동차보험 유무(0:없음/1:있음)
7. Vehicle_Age	자동차의 나이(1-2 Year/<1 Year/>2 Year)
8. Vehicle_Damage	자동차 사고 유무(Yes/No)
9. Annual_Premium	1년 간 지불하는 보험료
10. Policy_Sales_Channel	고객 채널 익명 코드 ie. 메일, 폰, 직접 방문 등
11. Vintage	보험 계약 일수
12. Response	자동차보험 관심 유무(0:없음/1:있음)

train.csv는 위의 12개의 변수로 이루어져 있다. 이때, 1, 3, 5, 9, 10, 11번 변수는 수치형 변수이고 2, 4, 6, 7, 8, 12번 변수는 범주형 변수이다. 특히 12번 변수 **Response**는 분석의 목표인 **반응변수**이다. 즉, 1-11번 변수를 이용해 12번을 설명하고 예측하는 모형을 만드는 것이 이번 분석의 목표이다.

이번 데이터셋에서 특이한 점은, 4번 변수 Driving_License는 0과 1을 가지는 이항변수이지만, 사실 모든 관측에서 1(면허 있음)로 나타난다. 또한 7번 변수 Vehicle_Age는 자동차의 나이를 나타내는 변수지만 양적 변수가 아니라 1년 이하, 1~2년 사이, 2년 이상으로 나뉘는 질적 변수이다. 이런 점 등을 주의하여 자세한 변수의 해석과 데이터의 분석은 분석 절차에서 진행하도록 하겠다.

1.4 분석 방법

이번 분석 모형의 반응변수 Response(자동차보험 관심 유무)는 0(관심 없음)과 1(관심 있음)로 이루어진 이항변수이기 때문에 분석 방법으로 회귀분석Ⅱ 시간에서 다룬 4가지 방법론(비선형/시계열/로지스틱/GLM) 중 **로지스틱 방법론**을 선택했다. 또한 **GLM**에서 반응변수 y를 이항분포로 가정하고, 연결함수를 로짓을 선택해도 로지스틱 분석이 가능하므로 이를 이용한 적합도 시도할 것이다. 즉, 두 가지 방법으로 로지스틱 모형의 적합을 할 예정이다. 이때 해당 데이터는 시간에 따라 관측된 자료가 아니므로 시계열 요소는 고려하지 않겠다.

분석 프로그램으로는 **Python**을 선택해 Python 내 두 가지 방법으로 로지스틱 모형을 적합 하고자 한다. 이때, 하나는 예측의 성능을 평가하는 데 이용하고 다른 하나는 로지스틱 모형을 해석하는 데 사용하겠다.

따라서 이번 분석에서 적합할 모형은 다음과 같다. 이때, $\pi(x)$는 반응변수 Response가 1일 확률이며, $x_1,\cdots,x_{11}$는 위에서 설명한 11개의 설명변수이다.

$$\log(\frac{\pi(x)}{1-\pi(x)}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_{11} x_{11} + \epsilon$$

1.5 결과 활용 및 기대 효과

해당 분석을 통해 실제로 자동차보험에 관심이 있다고 분류되는 고객에게 적극적으로 보험 가입을 홍보할 수 있을 것이며, 로지스틱 모형의 적합이 성공적일수록 더 적은 마케팅 비용으로 더 좋은 판매 성과를 올릴 수 있을 것이다. 이는 기존 고객에 대한 예측뿐만 아니라 각 변수가 자동차보험 가입에 어떤 영향을 미치는지를 적절히 해석한다면 새로 유입될 고객에게까지 확대해 적용될 수 있을 것이라 기대된다.

2. 본론

2.1 분석 방법 소개

앞서 언급했듯이, 이번 분석에서는 2가지 방법으로 로지스틱 회귀모형을 적합할 것이다. 이에 Python 프로그램을 이용해 다음 4단계의 분석 과정을 거치려고 한다.

Step1) 분석 준비하기 단계로 데이터 분석에 필요한 라이브러리와 데이터를 불러오는 가장 간단하고 기본적인 단계이다.

Step2) 데이터 살펴보기 단계로 분석 전 데이터가 잘 불러와 졌는지, 변수의 생김새는 어떤지 등 데이터의 기본 정보를 파악하고 분석의 감을 잡는 단계이다.

Step3) 데이터 전처리 단계로 데이터를 분석에 바로 사용할 수 있게 손보는 단계이다. 먼저 Encoding을 통해 범주형 변수를 적절히 코딩하고, 결측값이나 중복값이 있다면 처리한 뒤, 반응변수와 설명변수를 지정하고 train/test set으로 나누어 필요하다면 데이터 스케일링까지 진행할 것이다.

Step4) 로지스틱 모형 적합 단계로 Step1-3에서의 데이터 정보를 바탕으로 모형 적합을 할 것이다. 이때, **2가지 방법**의 로지스틱 모형 적합을 시도할 것이다. 첫 번째는 **sklearn 모듈의 LogisticRegression()**을 이용한 방법으로 분석의 최종 목표인 반응변수 예측의 관점에서 모형이 얼마나 적합 되었는지를 확인할 것이다. 즉, train set으로 모형을 적합하고 test set을 이용해 모형의 성능을 평가할 것이다.

다음으로는 **statmodels 패키지의 GLM**을 이용한 로지스틱 모형 적합을 통해 회귀분석II 시간에 배운 이론적 내용을 바탕으로 결과 해석을 중심으로 진행하겠다. 이때 반응변수 y의 분포는 이항분포를 가정하며, 연결함수로는 logit을 이용한다.

2.2 데이터 분석 및 결과 설명

2.2.1 분석 준비하기

(1) 필요한 라이브러리 불러오기

```
In [1]: import pandas  as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        import statsmodels.api as sm

        from sklearn.preprocessing import LabelEncoder
        from sklearn.model_selection import train_test_split
        from imblearn.over_sampling import RandomOverSampler
        from collections import Counter
        from sklearn.preprocessing import StandardScaler
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import precision_score,recall_score,accuracy_score,f1_score,c
        from sklearn.metrics import roc_curve
```

(2) 데이터 불러오기

```
In [2]: df=pd.read_csv("C:/Users/USER/Desktop/reg2_final/health_insurance.csv")
```

2.2.2 데이터 살펴보기

(1) 데이터 파악하기

● 상위 5개의 관측

```
In [3]: df.head()
```

Out[3]:

	id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Dam
0	1	Male	44	1	28.0	0	> 2 Years	
1	2	Male	76	1	3.0	0	1-2 Year	
2	3	Male	47	1	28.0	0	> 2 Years	
3	4	Male	21	1	11.0	1	< 1 Year	
4	5	Female	29	1	41.0	1	< 1 Year	

Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
> 2 Years	Yes	40454.0	26.0	217	1
1-2 Year	No	33536.0	26.0	183	0
> 2 Years	Yes	38294.0	26.0	27	1
< 1 Year	No	28619.0	152.0	203	0
< 1 Year	No	27496.0	152.0	39	0

⇒ 상위 5개의 관측을 통해 변수들의 대략적인 생김새를 파악할 수 있다.

⇒ id는 관측값의 순서를 나타내는 index의 역할만을 하는 변수로, 실제 분석에서는 제외하고 진행해도 될 것 같다.

⇒ Gender, Vehicle_Damage, Vehicle_Age는 범주형 변수로 분석 시 Encoding을 진행해야겠다.

⇒ 특히 Vehicle_Age는 수치형 변수일 것 같지만 범주형 변수임을 유의해야 한다.

● 데이터 정보

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 381109 entries, 0 to 381108
Data columns (total 12 columns):
id                    381109 non-null int64
Gender                381109 non-null object
Age                   381109 non-null int64
Driving_License       381109 non-null int64
Region_Code           381109 non-null float64
Previously_Insured    381109 non-null int64
Vehicle_Age           381109 non-null object
Vehicle_Damage        381109 non-null object
Annual_Premium        381109 non-null float64
Policy_Sales_Channel  381109 non-null float64
Vintage               381109 non-null int64
Response              381109 non-null int64
dtypes: float64(3), int64(6), object(3)
memory usage: 34.9+ MB
```

⇒ 381109개의 관측, 12개의 변수로 이루어진 데이터임을 확인할 수 있다.

● 요약통계량

```
In [5]: df.describe()
```

Out [5]:

	id	Age	Driving_License	Region_Code	Previously_Insured	Annual_
count	381109.000000	381109.000000	381109.000000	381109.000000	381109.000000	3811(
mean	190555.000000	38.822584	0.997869	26.388807	0.458210	3056
std	110016.836208	15.511611	0.046110	13.229888	0.498251	1721
min	1.000000	20.000000	0.000000	0.000000	0.000000	263
25%	95278.000000	25.000000	1.000000	15.000000	0.000000	244(
50%	190555.000000	36.000000	1.000000	28.000000	0.000000	3166
75%	285832.000000	49.000000	1.000000	35.000000	1.000000	394(
max	381109.000000	85.000000	1.000000	52.000000	1.000000	5401(

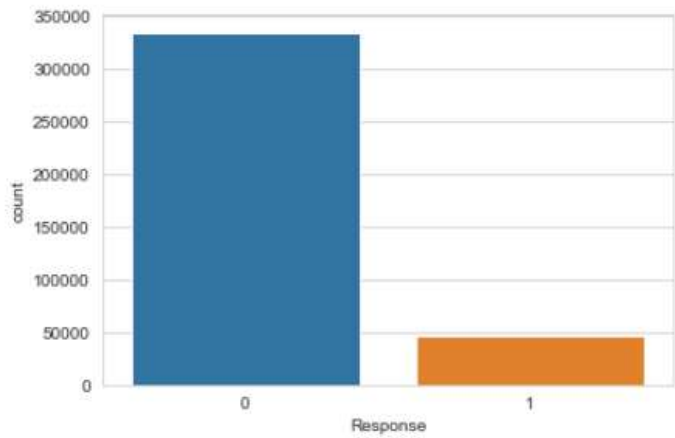
Previously_Insured	Annual_Premium	Policy_Sales_Channel	Vintage	Response
381109.000000	381109.000000	381109.000000	381109.000000	381109.000000
0.458210	30564.389581	112.034295	154.347397	0.122563
0.498251	17213.155057	54.203995	83.671304	0.327936
0.000000	2630.000000	1.000000	10.000000	0.000000
0.000000	24405.000000	29.000000	82.000000	0.000000
0.000000	31669.000000	133.000000	154.000000	0.000000
1.000000	39400.000000	152.000000	227.000000	0.000000
1.000000	540165.000000	163.000000	299.000000	1.000000

⇒ 수치형 변수들의 평균, 표준편차, 최댓값, 최솟값, 분위수 등의 요약통계량을 확인했다.
⇒ 이때, Driving_License, Previously_Insured, Response는 0과 1로 이루어진 이항변수이다.
⇒ 그러나, Driving_License의 통계량을 봤을 때, 평균이 0.998 정도로 거의 모든 값이 1로만 구성되어 있음을 볼 수 있다.

(2) 변수파악하기

● Response

```
In [6]: sns.set_style("whitegrid")
        sns.countplot(df['Response'],data=df)
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x147a9f56d30>
```



⇒ Response는 고객의 자동차보험 관심 유무를 나타내는 변수로, 이번 분석의 반응변수이다.
⇒ 0(관심 없음)이 1(관심 있음) 보다 월등히 높게 나타나고 있다.
⇒ 원활한 데이터 분석을 위해, 전처리 단계에서 불균형을 처리할 것이다.

● Gender, Driving_License, Previously_Insured, Vehicle_Damage

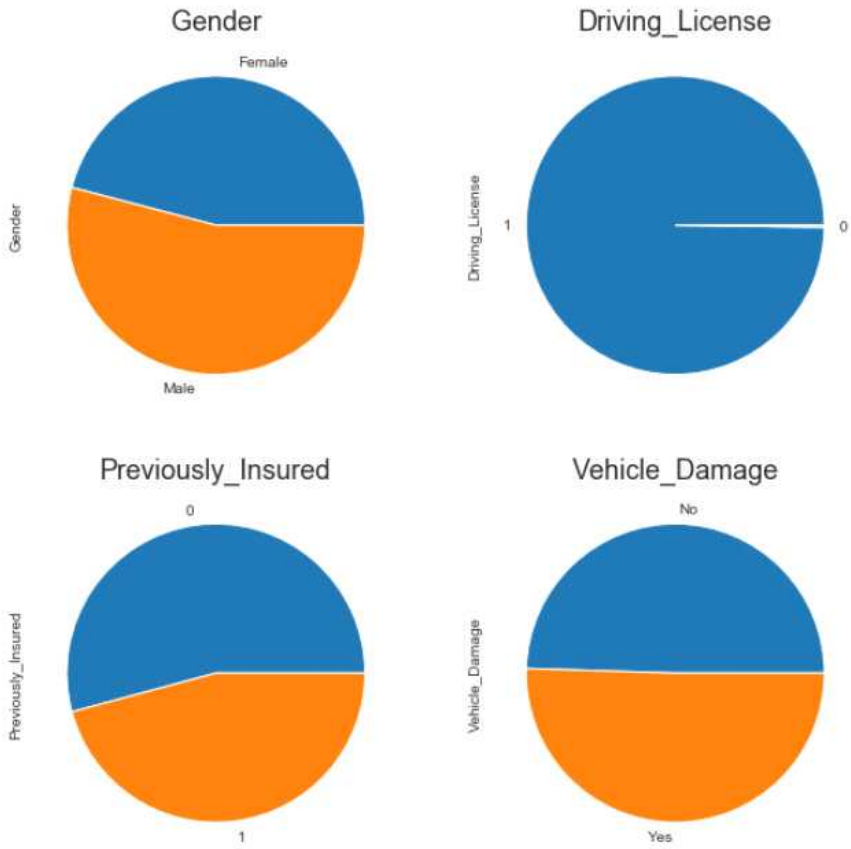
```
In [7]: fig, axarr = plt.subplots(2, 2, figsize=(10, 10))

        df['Gender'].value_counts().sort_index().plot.pie(
            ax=axarr[0][0])
        axarr[0][0].set_title("Gender", fontsize=18)
        df['Previously_Insured'].value_counts().sort_index().plot.pie(
            ax=axarr[1][0])
        axarr[1][0].set_title("Previously_Insured", fontsize=18)

        df['Vehicle_Damage'].value_counts().sort_index().plot.pie(
            ax=axarr[1][1])
        axarr[1][1].set_title("Vehicle_Damage", fontsize=18)

        df['Driving_License'].value_counts().head().plot.pie(
            ax=axarr[0][1])
        axarr[0][1].set_title("Driving_License", fontsize=18)
```

```
Out[7]: Text(0.5, 1.0, 'Driving_License')
```



⇒ 앞으로 살펴볼 4가지 이항변수에 대한 원그래프를 간단히 살펴보았다.

⇒ Gender는 남녀 성비가 거의 비슷하나 남성이 살짝 더 많은 것을 알 수 있다.

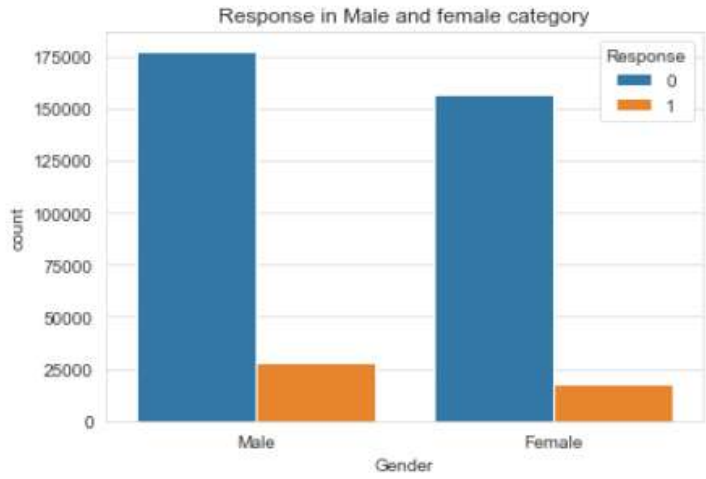
⇒ Driving_License는 모두 1(운전면허 있음)로 치우친 것이 눈에 띈다.

⇒ Previously_Insured는 거의 비슷하나 0(자동차보험 없음)이 살짝 더 많은 것을 알 수 있다.

⇒ Vehicle_Damage는 Yes/No가 매우 비슷한 비율임을 알 수 있다. 즉, 사고 경험의 여부는 거의 같다.

● Gender

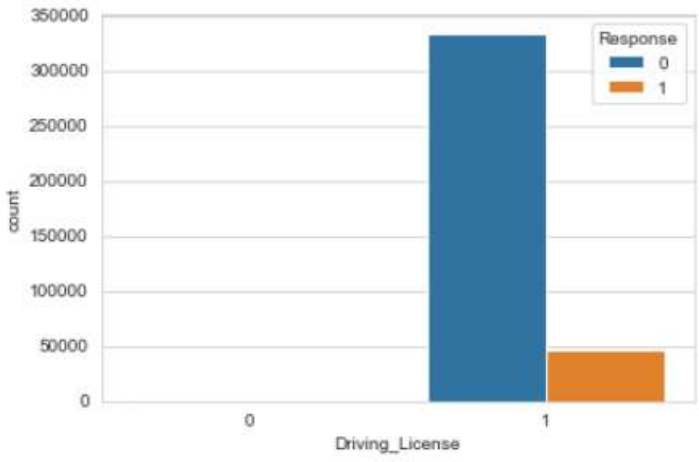
```
In [12]: sns.countplot(df['Gender'], hue = df['Response'])
         plt.title("Response in Male and female category")
         plt.show()
```



⇒ 먼저, 성별은 남성이 살짝 더 많지만 거의 고르게 분포하고 있었다.
⇒ Male일 때 Response가 1인 경우가 Female일 때보다 살짝 높게 나타난다.
⇒ 즉, 남성일 때 자동차보험에 관심 있을 확률이 살짝 높아 보인다.

● Driving_License

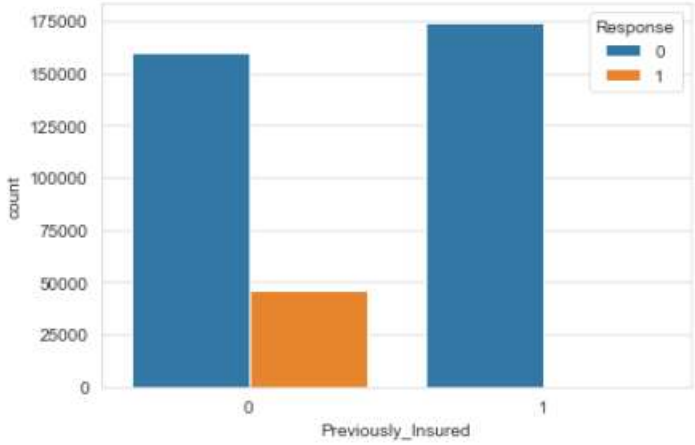
```
In [9]: sns.countplot(df['Driving_License'],hue=df['Response'])
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x147affab2e8>
```



⇒ 위에서 살펴봤듯이 Driving_License는 모두 1의 값을 가진다. 즉, 모두 면허를 가지고 있다.
⇒ 이는 보험회사 측에서 자동차보험의 관심 유무를 파악하기 위한 데이터이므로 애초에 운전면허가 있는 고객만 모았기 때문으로 추측한다.

● Previously_Insured

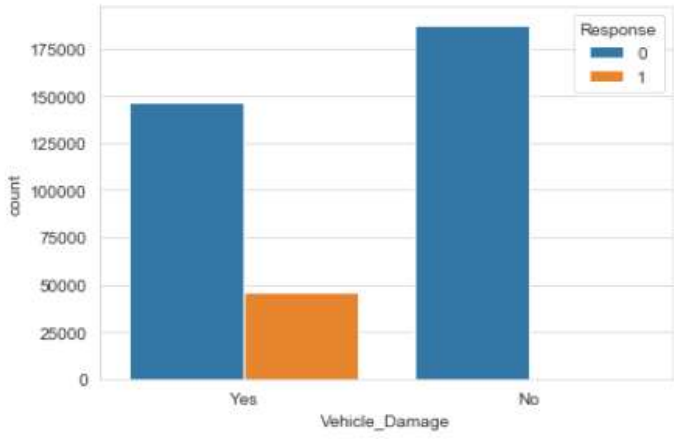
```
In [10]: sns.countplot(x='Previously_Insured',hue='Response',data=df)
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x147b0c7d160>
```



⇒ Previously_Insured에서 0은 자동차보험 없음, 1은 자동차보험 있음을 나타낸다.
⇒ 당연한 말인 것 같지만, 기존 자동차보험이 있는 고객은 새로 보험에 가입할 가능성이 낮은 듯하다.

● Vehicle_Damage

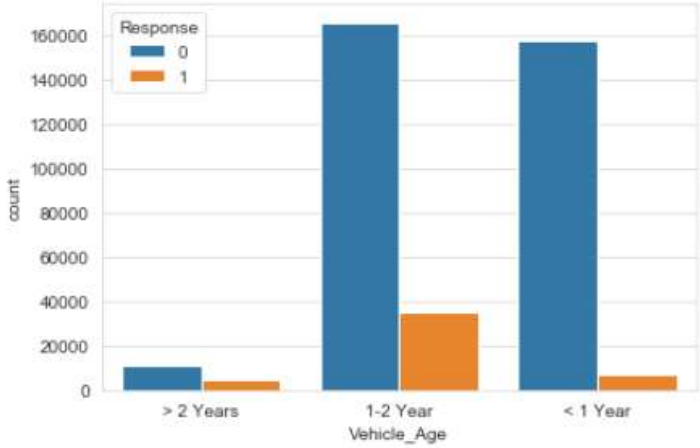
```
In [11]: sns.countplot(df['Vehicle_Damage'],hue=df['Response'])
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x147b0c7b978>
```



⇒ 위의 원그래프에서 Yes와 No의 비율이 거의 동일함을 확인했다.
⇒ 그래프 상으로는 자동차 사고 경험이 없으면 자동차보험에 관심을 가질 확률이 없는 것으로 나타난다.
⇒ 이는 실제 생활에 비교하기엔 극단적인 결과인 듯하다.

● Vehicle_Age

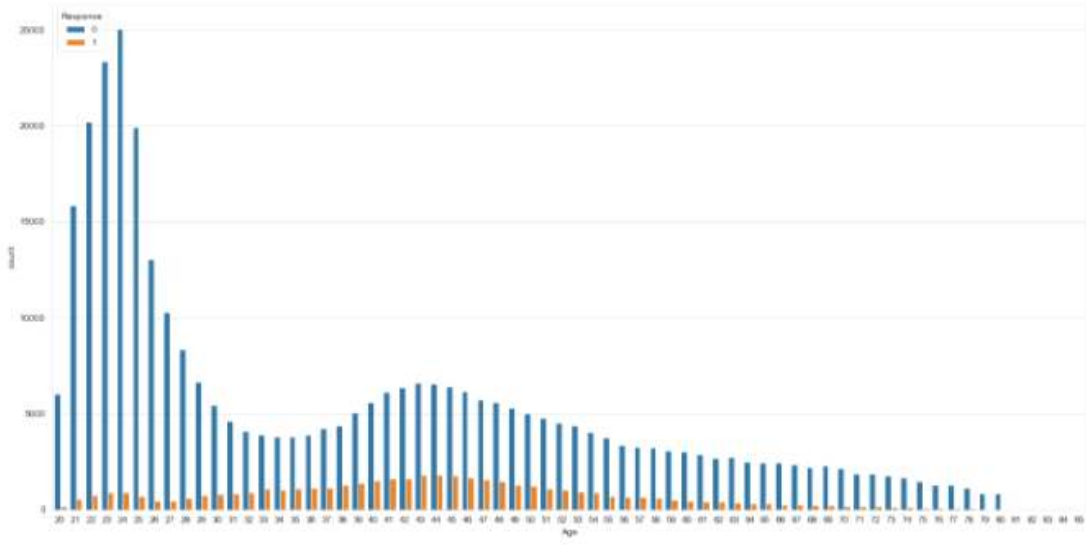
```
In [16]: sns.countplot(x='Vehicle_Age',hue='Response',data=df)
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x147b11df7b8>
```



⇒ Vehicle_Age 변수는 범주형 변수로 1년 이하, 1~2년, 2년 이상으로 나뉜다.
⇒ 1~2년 된 자동차를 가진 고객의 자동차보험 가입 확률이 가장 높게 나타나며, 2년 이상 된 자동차를 가진 고객은 가장 적은 관심을 보인다.
⇒ 그러나 이러한 결과는 애초에 1~2년 된 자동차의 관측이 가장 많았기 때문에 절대적인 해석은 아니다.

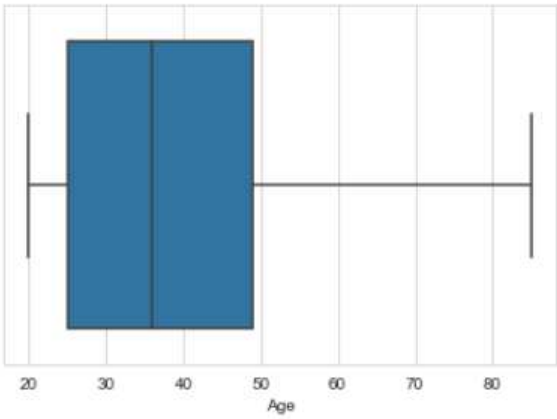
● Age

```
In [13]: plt.figure(figsize=(20,10))
         sns.countplot(x='Age',hue='Response',data=df)
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x147b0cc6080>
```



⇒ 나이의 분포는 20대 초중반이 가장 많으며, 30대보다 40대가 더 많이 관측되었다.
⇒ 30세 이하의 고객은 30~60세 고객보다 자동차보험에 관심이 없는 것으로 보이고, 30대 후반에서 40대 초반의 관측대비 관심이 가장 높은 듯하다.

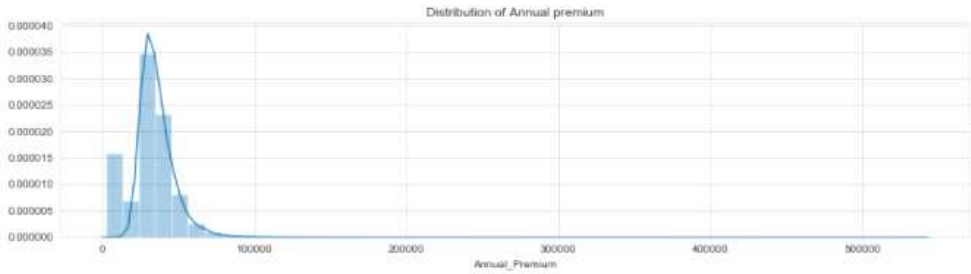
```
In [15]: sns.boxplot(df['Age'])
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x147b1190eb8>
```

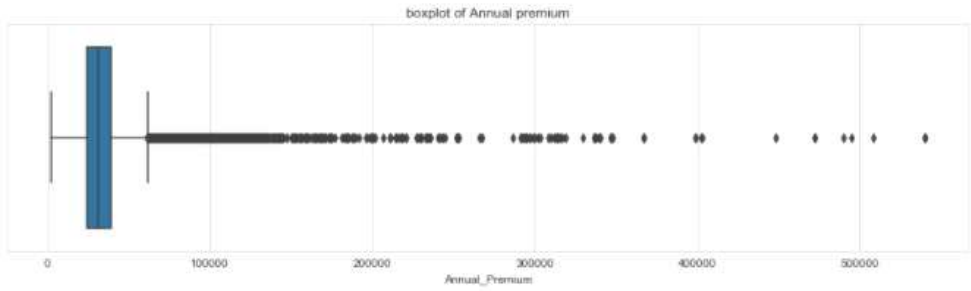


⇒ Boxplot을 보니 이상치는 없는 듯하다.

● Annual_Premium

```
In [17]: plt.figure(figsize=(15,8))
         plt.subplot(2,1,1)
         sns.distplot(df['Annual_Premium'])
         plt.title("Distribution of Annual premium")
         plt.show()
         plt.figure(figsize=(15,8))
         plt.subplot(2,1,2)
         sns.boxplot(df['Annual_Premium'])
         plt.title("boxplot of Annual premium")
         plt.show()
```

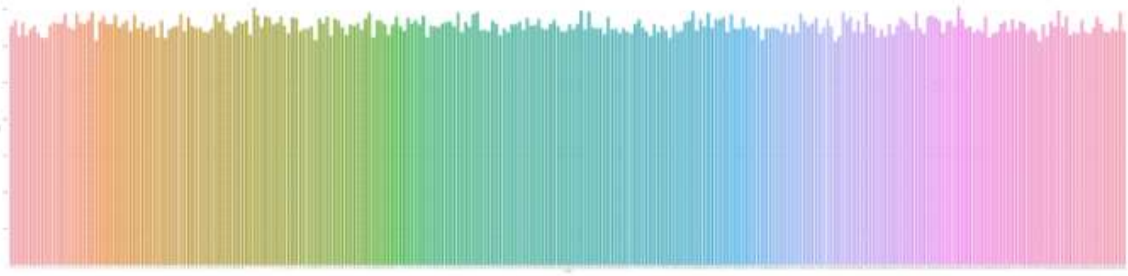




⇒ 연간 보험료의 분포는 왼쪽으로 쏠린 근사적인 정규분포 형태임을 확인할 수 있다.
⇒ Boxplot을 보니 많은 이상치가 존재한다.
⇒ 즉 대부분 고객은 비슷한 수준의 보험료를 지불하지만, 고액의 보험료를 지불하는 고객도 꽤 존재하는 듯하다.

● Vintage

```
In [24]: fig_dims = (80, 20)
         fig, ax = plt.subplots(figsize=fig_dims)
         sns.countplot('Vintage', data = df, ax = ax)
         ax.set(xlabel='Vintage', ylabel='Count')
         plt.show()
```



⇒ Vintage는 고객의 보험 계약 일수를 나타내는 변수로, 고르게 분포하고 있음을 확인할 수 있다.

● 그 외: id, Region_Code, Policy_Sales_Channel

⇒ 나머지 3개의 변수는 단순 고객 번호를 나타내는 id이거나 코드를 나타내는 숫자이므로 생략하겠다.

2.2.3 데이터 전처리

(1) Encoding

● 데이터 변환

```
In [28]: df['Gender'].replace(to_replace={'Male':0,'Female':1},
                              inplace=True)
         df['Vehicle_Damage'].replace(to_replace={'No':0,'Yes':1},
                              inplace=True)
         df['Vehicle_Age'].replace(to_replace={'< 1 Year':0,'1-2 Year':1,'> 2 Years':2},
                              inplace=True)
```

⇒ 분석을 위해 범주형 변수 Gender, Vehicle_Age, Vehicle_Damage를 Encoding 했다.

⇒ Gender의 Male은 0, Female은 1의 값으로 대체했다.

⇒ Vehicle_Damage의 No는 0, Yes는 1의 값으로 대체했다.

⇒ Vehicle_Age의 <1 Year는 0, 1-2 Year는 1, >2 Year는 2로 대체했다.

● 변환된 데이터 확인

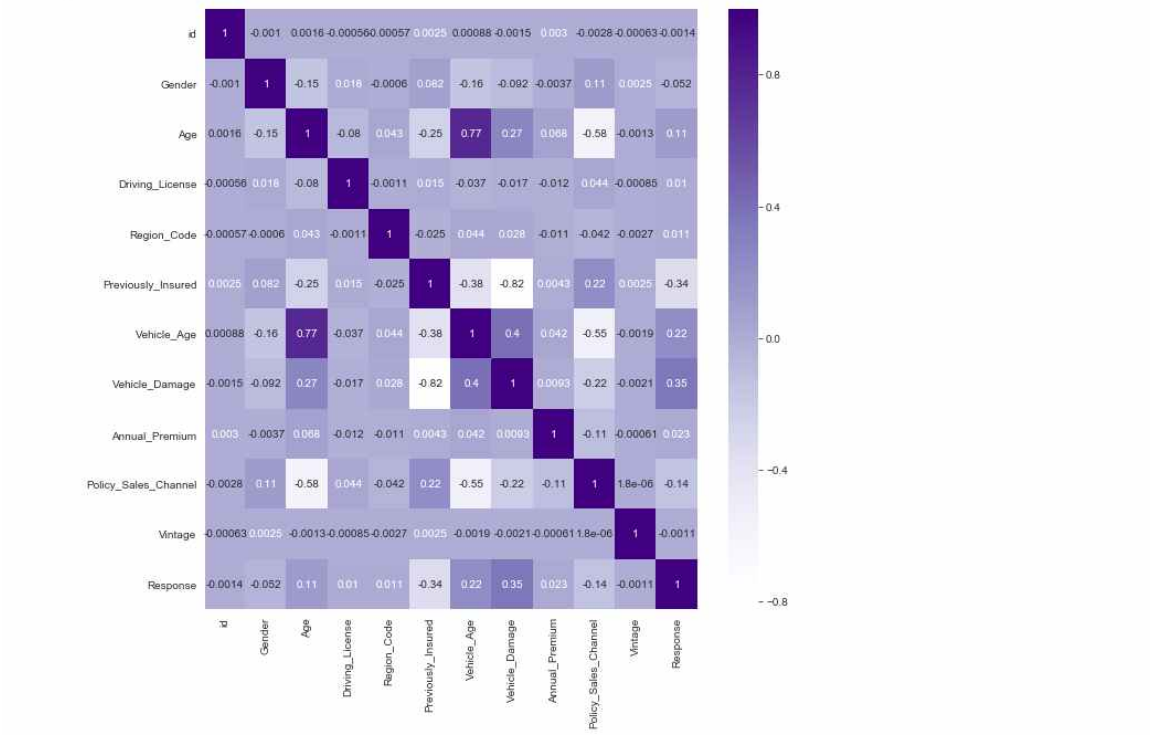
```
In [29]: df.dtypes
```

```
Out[29]: id                      int64
         Gender                  int64
         Age                     int64
         Driving_License         int64
         Region_Code           float64
         Previously_Insured      int64
         Vehicle_Age             int64
         Vehicle_Damage          int64
         Annual_Premium        float64
         Policy_Sales_Channel  float64
         Vintage                 int64
         Response                int64
         dtype: object
```

⇒ 더 이상 object타입의 변수가 존재하지 않음을 확인할 수 있다. 즉, 변환이 잘 되었다.

● Correlation Heatmap

```
In [32]: plt.figure(figsize=(10,10))
         cor=df.corr()
         sns.heatmap(cor,annot=True,cmap=plt.cm.Purples)
         plt.show()
```



⇒ 변수 간의 상관관계를 확인 결과, 반응변수와 가장 큰 관계가 있는 변수는 Vehicle_Damage 이며 가장 작은 관계가 있는 변수는 Vintage임을 알 수 있다.
⇒ 그 외에도 id 변수의 상관관계도 낮게 나온 것을 보아 처음 생각했던 대로 제거하고 진행해도 될 것 같다.

(2) 결측값 및 중복값 처리

● 결측값

```
In [75]: df.isna().sum()
```

```
Out[75]: id                    0
         Gender                0
         Age                   0
         Driving_License       0
         Region_Code           0
         Previously_Insured    0
         Vehicle_Age           0
         Vehicle_Damage        0
         Annual_Premium        0
         Policy_Sales_Channel  0
         Vintage               0
         Response              0
         dtype: int64
```

⇒ 결측값이 존재하지 않기 때문에 결측값 처리는 생략하겠다.

● 중복값

```
In [77]: duplicate=df[df.duplicated()]
         print(duplicate)
```

```
Empty DataFrame
Columns: [id, Gender, Age, Driving_License, Region_Code, Previously_Insured, Vehicle
_Age, Vehicle_Damage, Annual_Premium, Policy_Sales_Channel, Vintage, Response]
Index: []
```

⇒ 중복값 또한 존재하지 않기 때문에 중복값 처리는 생략하겠다.

(3) 변수 준비하기

● 반응변수와 설명변수로 분리

```
In [36]: df=df.drop(columns=['id'])
```

```
In [37]: y=df.Response
         x=df.drop(columns=['Response'])
```

⇒ 위에서 봤듯이, id는 인덱스 역할을 하므로 설명변수로 부적절하다고 판단했으며, 상관계수를 봐도 큰 관련성이 없음을 확인했다. 따라서 제거하고 진행하겠다.
⇒ 그 외의 변수는 제거하지 않고 반응변수는 Response, 설명변수는 나머지 10개의 변수로 설정했다.

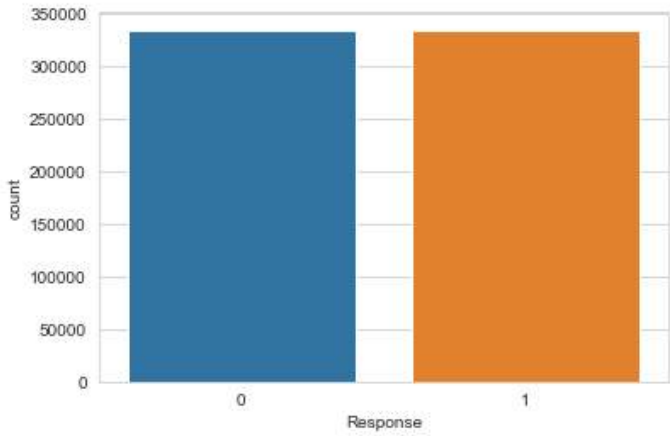
● 불균형 데이터 처리

```
In [38]: randomsample= RandomOverSampler()
x_new,y_new=randomsample.fit_sample(x,y)

print('Original dataset shape {}'.format(Counter(y)))
print('Resampled dataset shape {}'.format(Counter(y_new)))
sns.countplot(y_new)
```

```
Original dataset shape Counter({0: 334399, 1: 46710})
Resampled dataset shape Counter({1: 334399, 0: 334399})
```

```
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x147b1477198>
```



⇒ 위에서 살펴봤듯이 Response가 한 클래스로 치우친 클래스 불균형이 존재했으므로 이를 해결하기 위해 재샘플링 기법을 사용했다.
⇒ 성공적으로 클래스가 분배되었음을 확인할 수 있다.

(4) 데이터 나누기 – train/test set

```
In [39]: xtrain,xtest,ytrain,ytest=train_test_split(x_new,y_new,test_size=.3,random_state=42)
```

⇒ 데이터를 train set(70%)과 test set(30%)로 나누었다.
⇒ 따라서 train set을 이용해서 모형을 적합하고 test set을 이용해서 모형의 성능을 평가하겠다.

(5) 데이터 스케일링

```
In [40]: scaler=StandardScaler()
         xtrain=scaler.fit_transform(xtrain)
         xtest=scaler.transform(xtest)
```

⇒ 이제 데이터 스케일링을 끝으로 데이터 전처리 단계를 마무리하고 모형적합을 시작하겠다.

2.2.4 로지스틱 모형 적합

(1) 방법1-로지스틱: sklearn 모듈의 LogisticRegression()을 이용하여 로지스틱 모형 적합

● 모형적합

```
In [47]: lr=LogisticRegression()
         lr=lr.fit(xtrain, ytrain)
         print("절편:",lr.intercept_)
         print("회귀계수:", lr.coef_)
```

```
절편: [-0.82979129]
회귀계수: [[-0.05643861 -0.29452667  0.05334393 -0.00424658 -1.75365562  0.42674385
   0.91836644  0.01624181 -0.14766875 -0.00430147]]
```

⇒ 먼저 train set을 이용해 로지스틱 모형을 적합한 결과는 다음과 같다.

⇒ 로지스틱 모형: $\log(\frac{\pi(x)}{1-\pi(x)}) = -0.83 - 0.056x_1 - 0.295x_2 + 0.053x_3 - 0.004x_4 - 1.754x_5 + 0.427x_6 + 0.918x_7 + 0.016x_8 - 0.148x_9 - 0.004x_{10}$

⇒ 성공확률: $\pi(x) = \frac{1}{1+e^{-(-0.83 - 0.056x_1 - 0.295x_2 + 0.053x_3 - 0.004x_4 - 1.754x_5 + 0.427x_6 + 0.918x_7 + 0.016x_8 - 0.148x_9 - 0.004x_{10})}}$

⇒ 이때 회귀계수의 모수 추정은 최대가능도추정법을 이용하며, 적합 된 로지스틱 모형의 절편과 회귀계수 해석은 방법2에서 할 것이므로 여기선 간단히 확인만 하고 넘어가겠다.

● 예측

```
In [43]: pred=lr.predict(xtest)
         print("Logistic regression Reports:₩n", classification_report(ytest, pred))
```

```
Logistic regression Reports:
              precision    recall  f1-score   support

           0       0.96      0.60      0.74    100136
           1       0.71      0.98      0.82    100504

    accuracy                           0.79    200640
   macro avg       0.83      0.79      0.78    200640
weighted avg       0.83      0.79      0.78    200640
```

⇒ 다음으로, test set을 이용해 예측한 결과는 위와 같다.
⇒ 모델 성능 평가 지표 확인 결과, 정밀도는 0.83, 재현율은 0.79로 나타났다.
⇒ 이때, 정밀도(Precision)는 Confusion Matrix에서 TP/(TP+FP)로 구하며, True로 예측한 관측값 중 실제값이 True인 것의 비율을 나타낸다.
⇒ 또한 재현율(Recall)은 Confusion Matrix에서 TP/(TP+FN)으로 구하며, 실제값이 True인 관측값 중 예측도 True인 것의 비율을 나타낸다.
⇒ 정밀도와 재현율의 가중치에 따라 해석이 달라질 수 있는데, 이때 둘을 동일한 가중치로 평균을 낸 값이 f1-score이며 2*Precision*Recall/(Precision+Recall)로 구할 수 있다. 이 경우 0.78로 나타난다.
⇒ 평가 지표를 살펴본 결과, 전체적으로 모델 성능이 나쁘지 않은 것 같다.

● Accuracy

```
In [44]: acc=accuracy_score(ytest,pred)
         print("Accuracy of Logistic Regression" + ' : ' + str(acc))

         Accuracy of Logistic Regression : 0.7862190988835726
```

⇒ 모델의 정확도(Accuracy)는 Confusion Matrix에서 (TP+TN)/(TP+TN+FP+FN)으로 구하며, 전체 관측값 중 실체값과 예측값이 일치한 비율이다.
⇒ 계산 결과, 정확도는 약 79%로 이는 100개 중 79개는 예측에 성공, 21개는 예측에 실패했다는 뜻이다. 아주 우수하진 않지만, 나쁘진 않은 수치인 듯하다.

● AUC

```
In [45]: AUC=roc_auc_score(pred,ytest)
         print("ROC_AUC Score of Logistic Regression:" + ' : ' + str(AUC))

         ROC_AUC Score of Logistic Regression: : 0.8344759356977003
```

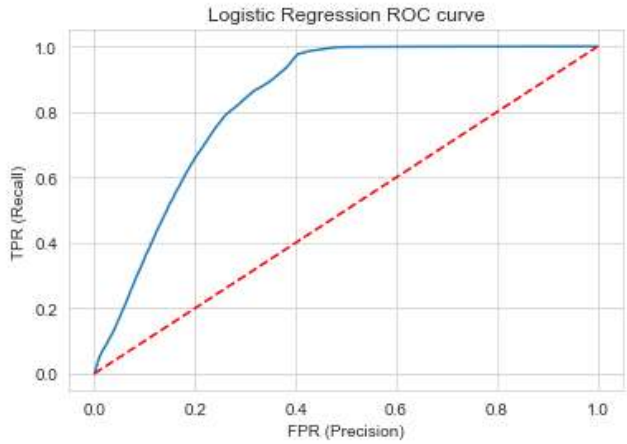
⇒ AUC는 ROC Curve 아래의 면적으로 0과 1사이의 값을 가지며 1에 가까울수록 좋은 모형이다.
⇒ 계산 결과, AUC는 약 0.83으로 역시 괜찮은 결과인 듯하다.

● ROC curve

```
In [50]: lr_probability = lr.predict_proba(xtest)[:,1]
         fpr, tpr, _ = roc_curve(ytest, lr_probability)

         plt.title('Logistic Regression ROC curve')
         plt.xlabel('FPR (Precision)')
         plt.ylabel('TPR (Recall)')

         plt.plot(fpr,tpr)
         plt.plot((0,1), ls='dashed',color='red')
         plt.show()
```



⇒ ROC Curve의 x축은 (1-특이도), y축은 민감도로 그래프가 (0, 1)에 가까이 갈수록 우수한 모형이다.

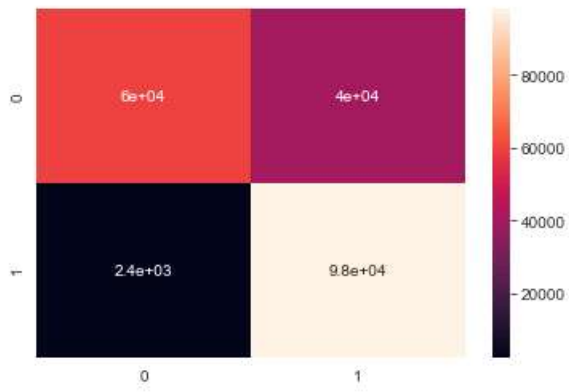
⇒ 다른 방법의 모형 적합을 해보지 않았기 때문에 가장 좋은 분류 방법일 것이라고 확신할 수는 없지만, 로지스틱 모형이 나쁘진 않은 결과를 내는 것 같다.

● Confusion Matrix

```
In [48]: cm=confusion_matrix(ytest,pred)
         print(cm)
         sns.heatmap(cm,annot=True)

[[59660 40476]
 [ 2417 98087]]

Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x147b197ca58>
```



⇒ 참고로 Confusion Matrix를 확인하자면 위 히트맵의 빨강색, 자주색, 검정색, 살구색은 각각 TP, FN, FP, TN을 의미한다.

(2) 방법2-GLM: statmodels 패키지의 GLM을 이용하여 로지스틱 모형 적합

● 모형적합

```
In [35]: logit = sm.GLM(ytrain, xtrain, family=sm.families.Binomial())
         result = logit.fit(method="newton")
```

⇒ π: $E[y]$으로, $\Pr(Y=1)$의 값과 같으며 성공확률을 의미한다. 즉, 이번 데이터의 경우에는 고객이 자동차보험에 관심을 보일 확률을 나타낸다.

⇒ $\frac{\pi}{1-\pi}$: 성공확률(π)과 실패할 확률($1-\pi$)의 비로 Odds의 정의이다.

⇒ $\log(\frac{\pi}{1-\pi})$: Odds에 로그를 취한 값으로 로그오즈, 로짓이라고도 한다.

⇒ $\log(\frac{\pi(x)}{1-\pi(x)}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$: 우리의 분석에서 최종적으로 얻고자 하는 로지스틱 모형이다.

⇒ 회귀계수의 추정 과정 중 비선형을 해결하기 위한 반복법으로 뉴턴법을 이용했다.

● 모형적합 결과

```
In [34]: result.summary2()
```

Out [34]:

Model:	GLM	Method:	IRLS
Link Function:	logit	AIC:	430993.3089
Dependent Variable:	Response	BIC:	-5681429.6685
Date:	2020-12-09 00:45	Log-Likelihood:	-2.1549e+05
No. Observations:	468158	LL-Null:	-3.2450e+05
Df Model:	9	Deviance:	4.3097e+05
Df Residuals:	468148	Pearson chi2:	4.08e+05
Converged:	1.0000	Scale:	1.0000
No. Iterations:	3.0000		

⇒ GLM에서 확률적 성분인 반응변수는 이항분포를 따르고, 체계적 성분인 설명변수는 모수의 선형결합으로 표현하였으며 연결함수는 로짓 Link를 이용해서 로지스틱 모형을 적합했다.

⇒ 모수 추정 방법은 IRLS를 이용했으며 뉴턴법을 통한 반복법을 이용해 반복수 3번 만에 수

렴했음을 확인할 수 있다.

⇒ Pseudo R-square 값이 0.336으로 낮게 나왔지만 로지스틱 회귀분석에서 보통 결정계수는 낮게 나오므로 넘어갈 수 있다. 하지만 그 외 AIC, BIC, Deviance, Pearson chi-square 통계량 등을 보면 우수한 모형은 아닌 하다. 더 나은 모형 적합을 위해서는 변수를 선택하는 과정이 필요할 것 같다.

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
x1	-0.0493	0.0038	-12.8932	0.0000	-0.0568	-0.0418
x2	-0.3223	0.0055	-58.7185	0.0000	-0.3331	-0.3116
x3	0.0523	0.0035	15.0123	0.0000	0.0455	0.0592
x4	-0.0061	0.0038	-1.5973	0.1102	-0.0137	0.0014
x5	-0.9425	0.0069	-136.7617	0.0000	-0.9560	-0.9290
x6	0.4658	0.0058	80.5390	0.0000	0.4545	0.4771
x7	0.7751	0.0064	121.3530	0.0000	0.7625	0.7876
x8	0.0261	0.0038	6.8453	0.0000	0.0186	0.0335
x9	-0.1464	0.0044	-33.0905	0.0000	-0.1551	-0.1377
x10	-0.0029	0.0038	-0.7704	0.4411	-0.0103	0.0045

⇒ 로지스틱 모형: $\log(\frac{\pi(x)}{1-\pi(x)}) = -0.0493x_1 - 0.3223x_2 + 0.0523x_3 - 0.9425x_5 + 0.4658x_6 + 0.7751x_7 + 0.0261x_8 - 0.1464x_9$

⇒ 성공확률: $\pi(x) = \frac{1}{1+e^{-(-0.0493x_1 - 0.3223x_2 + 0.0523x_3 - 0.9425x_5 + 0.4658x_6 + 0.7516x_7 + 0.0261x_8 - 0.1464x_9)}}$

⇒ x4(Region_Code)의 95% 신뢰구간이 [-0.0137, 0.0014]로 0을 포함하며, 유의확률은 0.1102으로 유의수준 0.05에서 기각하겠다.

⇒ 마찬가지로 x10(Vintage)의 95% 신뢰구간이 [-0.0103, 0.0045]로 0을 포함하며, 유의확률은 0.4411로 유의수준 0.05에서 기각하겠다.

⇒ 회귀계수의 부호가 양수면 해당 변수가 반응변수가 1일 확률(π)이 상승하는 방향으로 영향을 미친다는 것이고, 반대로 음수면 반응변수가 0일 확률($1-\pi$)이 상승하는 방향으로 영향을 미친다고 해석할 수 있다.

⇒즉, 부호가 양수인 x3(Driving_License), x6(Vehicle_Age), x7(Vehicle_Damage), x8(Annual_Premium)은 고객의 자동차보험 관심을 높이는 요인이며 부호가 음수인 x1(Gender), x2(Age), x5(Previously_Insured), x9(Policy_Sales_Channel)는 자동차보험의 관심을 줄이는 요인이라고 해석할 수 있다.

⇒ 각 변수가 반응변수에 영향을 미치는 정도는 오즈비를 통해서 파악할 수 있으므로 이어서 확인하겠다.

● 오즈비

```
In [36]: np.exp(result.params)

Out[36]: x1      0.951868
         x2      0.724462
         x3      1.053714
         x4      0.993876
         x5      0.389645
         x6      1.593275
         x7      2.170713
         x8      1.026404
         x9      0.863823
         x10     0.997085
         dtype: float64
```

⇒ 오즈비는 1을 기준으로 해석할 수 있는데, 1에 가까울수록 영향이 미미하며 1에서 멀어질수록 영향이 크다. 이때 1보다 작으면 Y=1의 확률이 낮아지는 쪽으로 영향을 미치며, 1보다 크면 Y=1의 확률이 커지는 쪽으로 영향을 미친다고 해석할 수 있다.
⇒ x1(Gender): 성별이 남성일 때보다 여성일 때 자동차보험에 관심을 가질 오즈가 약 0.95배 증가한다. 이는 1에 가까운 값으로 성별은 거의 영향을 미치지 않는 듯하다.
⇒ x2(Age): 나이가 한 단위 증가하면 자동차보험에 관심을 가질 오즈가 약 0.72배 증가한다.
⇒ x3(Driving_License): 운전면허가 없을 때보다 있을 때 자동차보험에 관심을 가질 오즈가 약 1.05배 증가한다. 역시 1에 가까운 값으로 거의 영향을 미치지 않는다.
⇒ x5(Previously_Insured): 기존 보험이 없는 고객보다 있는 고객이 자동차보험에 관심을 가질 오즈가 약 0.39배 증가한다.
⇒ x6(Vehicle_Age): 자동차의 나이가 1년 이하, 1~2년, 2년 이상으로 수준이 변할 때 자동차보험에 관심을 가질 오즈가 약 1.59배 증가한다.
⇒ x7(Vehicle_Damage): 자동차 사고 경험이 없는 고객보다 있는 고객이 자동차보험에 관심을 가질 오즈가 2.17배 증가한다. 이는 가장 큰 영향을 미치는 변수이다.
⇒ x8(Annual_Premium): 연간 보험료가 한 단위 증가할 때 자동차보험에 관심을 가질 오즈가 1.02배 증가한다. 역시 1에 가까운 값이므로 거의 영향을 미치지 않는다.
⇒ x9(Policy_Sales_Channel): 고객 채널 익명 코드가 한 단위 증가할 때 자동차보험에 관심을 가질 오즈가 0.86배 증가한다.

2.3 결과 해석 및 분석의 타당성 설명

분석 결과 중 먼저 두 번째 방법으로 적합한 로지스틱 모형의 해석부터 정리하자면 고객이 자동차보험에 관심을 가지는 것에 **긍정적으로 작용하는 변수**는 **자동차의 나이와 사고 경험**이었다. 이때 자동차의 나이가 많을 때, 사고 경험이 있을 때 자동차보험에 대한 관심이 커진다.

이는 실생활에 대입해서 생각해봐도 충분히 유추할 만한 결과였다. 자동차가 오래될수록 사고의 위험에 경각심을 가질 수 있으며 사고 경험이 있을 때 사고의 무서움을 알고 이에 대비하기 위한 보험을 충분히 찾을 만하다.

반면 고객이 자동차보험에 관심을 가지는 데 **부정적으로 작용하는 변수**는 **고객의 나이와 기존 보험 유무, 고객 채널 익명 코드**로 나타난다. 나이가 많을수록 자동차보험에 회의적이며 기존 보험을 가지고 있을 때는 자동차보험에 관심이 없는 것으로 나타난다. 이는 20대 초중반의 운전을 막 시작한 이들은 사고의 위험이 더 크기 때문에 보험에도 더 관심을 가질 것으로 추론할 수 있으며, 당연히 기존에 가입한 보험이 있는 경우 추가로 보험에 관심을 가질 확률이 낮을 것이라고도 생각할 수 있다. 다만 고객 채널 익명 코드의 경우는 비록 유의하게 나왔으나 지역코드나 ID와 같이 수치형 변수임에도 수치 자체가 어떤 의미를 지니는 게 아닌, 코드화된 값이기 때문에 사실 반응변수로는 적절치 못한 것 같아서 애초에 제외하고 진행해도 될 것 같다. 실제로 지역코드는 유의확률에 의해 기각되었다.

그 외, 성별이나 운전면허 유무, 지역 코드, 연간 보험료 등은 자동차보험 가입 여부에 큰 영향이 없는 것으로 나타났다. 성별의 경우 큰 차이가 없을 것 같긴 하지만 아무래도 여성의 경우가 좀 더 관심을 보일 것이라 예상한 것과는 다른 결과였고, 연간 보험료도 큰 의미가 없다고 나온 건 의외였다. 아무래도 연간 지불하는 보험료가 많으면 추가로 보험료를 내는 것을 꺼려서 가입을 덜 하거나, 혹은 많은 돈을 낼 수 있는 능력이 있다는 뜻이므로 추가로도 가입할 확률이 높을 거라는 결과를 예상했었다.

마지막으로 첫 번째 방법에서 적합한 **로지스틱 예측 모델의 성능**은 **Accuracy는 약 79%, AUC는 약 83%**로 나쁘지 않은 결과를 보여줬다. ROC Curve를 봐도 꽤 괜찮은 모양을 가지고 있었다. 그러나 다른 분류 기법을 통한 더 나은 모형 구축의 가능성은 배제할 수 없을 듯하다.

3. 결론

3.1 분석 결과 요약

이번 연구의 목표는 고객의 자동차보험에 대한 관심을 나타내는 로지스틱 예측 모형을 적합하고 결과를 이해하는 것이었다. 모형 적합 결과는 다음과 같이 정리할 수 있다. 이때 두껍게 표시한 변수는 그중 가장 큰 영향을 준다는 의미이다.

● 로지스틱 분류 모형의 성능

- 정밀도(Precision): 약 83%
- 재현율(Recall): 약 79%
- 정확도(Accuracy): 약 79%
- AUC: 약 83%

● 로지스틱 모형

$$\log(\frac{\pi(x)}{1-\pi(x)}) = -0.0493x_1 - 0.3223x_2 + 0.0523x_3 - 0.9425x_5 + 0.4658x_6 + 0.7751x_7 + 0.0261x_8 - 0.1464x_9$$

● 긍정적인 영향을 미치는 변수

: 자동차의 나이(Vehicle_Age), **자동차사고 경험(Vehicle_Damage)**

● 부정적인 영향을 미치는 변수

: 고객의 나이(Age), **기존보험 유무(Previously_Insured)**, 고객채널익명코드(Policy_Sales_Channel)

● 큰 영향을 미치지 않는 변수

: 고격의 성별(Gender), 운전면허 유무(Driving_License), 지역코드(Region_Code),
고객의 계약일수(Vintage), 연간 보험료(Annual_Premium)

3.2 분석의 장점 및 한계점 설명

로지스틱 회귀모형을 이용해 간단하게 모형을 적합한 결과, 괜찮은 예측 성능을 얻었으며 편리한 해석이 가능했다. 그러나 이번 분석에서는 변수 선택의 과정을 거치지 않고 임의로 제외한 id 변수 외에는 모든 변수를 사용해 모형을 적합했으므로 로지스틱 모형 내에서도 향상의 여지가 남아있다. 변수 선택법에 의해서 모형을 다시 적합하거나 혹은 유의확률에 의해 기각된 Region_Code와 Vintage 변수를 제외하고도 Person_Sales_Channel 같은 단순 코드를 나타내는 변수는 제외하고 진행해보는 것도 괜찮을 듯하다. 또한 최적의 모델을 찾기 위해서는 로지스틱 모형 외에도 다양한 분류 기법을 이용하고 비교해보는 것이 좋을 것이다.

3.3 추가 연구사항 제안

해당 데이터는 건강보험회사 고객의 자동차보험에 대한 관심 유무에 대한 가장 적절한 예측 모형을 찾기 위해서 제공되었다. 이에 이번 분석에서는 로지스틱 회귀모형을 이용한 적합을 시도했지만, 더 나은 분류 및 예측 모델을 찾기 위해 랜덤포레스트, KNN, 부스팅 등의 다양한 방법을 이용할 수 있을 것이다.

또한 단순히 기존 고객에 대한 예측의 과제에서 벗어나 본질적으로 자동차보험을 많이 팔 수 있는 전략적인 시점으로 접근한다면, 이에 영향을 미치는 변수를 파악하는 추가적인 자료와 분석이 필요할 것이다. 즉, 해당 상품에 관심이 있는 것과 실제로 해당 상품을 구매할지는 차이가 있으므로 단순히 고객의 정보를 이용한 예측에서 끝나지 않고, 자동차보험 마케팅의 본질을 이해하고 실질적인 구매로 이루어지는 것에 대한 분석으로 확장되어야 한다는 것이다. 이번엔 기존 건강보험 고객의 정보를 가지고 모델을 적합했다면, 다음에는 실제로 자동차보험에 가입한 고객의 정보 등의 분석으로 이어지면 좋을 것 같다.

참고문헌

데이터 출처
https://www.kaggle.com/anmolkumar/health-insurance-cross-sell-prediction

관련 문헌 1. (논문) 자동차보험과 산업재해보상보험과의 관계에 대한 연구
http://www.riss.kr/search/detail/DetailView.do?p_mat_type=be54d9b8bc7cdb09&control_no=c8bb0e69edf6512fffe0bdc3ef48d419

관련 문헌 2. (논문) 자동차보험의 마케팅전략에 관한 연구
http://www.riss.kr/search/detail/DetailView.do?p_mat_type=be54d9b8bc7cdb09&control_no=02e89f5afcb5fe44

관련 문헌 3. (분석) Vehicle Insurance EDA and boosting models
https://www.kaggle.com/yashvi/vehicle-insurance-eda-and-boosting-models