랜덤 포레스트와 상관 관계 검정을 통한
자동차보험 손해액에 영향을 미치는 주요 변인 연구

초록버블티

김보경 홍지은

< 목차 >

I. 서론

II. 본론

1. 해당 연구에 사용된 기법

2. 연구에 사용된 자료와 파생변수

3. 연구 과정

A. 모델을 통한 주요 변수 채택

B. X축(주요 설명변수)과 Y축(종속변수) 자료의 시각화

C. 시각화에서 얻은 인사이트를 이용한 통계적 검정

4. 논리적 관계 도출

5. 실현 가능한 방안

III. 결론

I. 서론

손해보험은 생명보험과 마찬가지로 다수의 보험계약에 의한 대수의 법칙이 작용되어 이론적으로 위험이 평준화된다. 하지만 손해보험회사가 담보하는 위험은 발생의 확률이 불규칙적이고 때로는 큰 화재, 태풍, 지진 등으로 인하여 예상치 못한 거대한 위험이 발생하기도 한다. 위의 상황이 발생했을 때 지급해야 하는 거액의 보험금은 책임준비금만으로는 충당할 수 없다. 따라서 예상 사고율을 초과하는 거대 위험에 대비하기 위해 일정금액을 책임준비금에 추가하여 비상위험준비금으로 적립한다. 이 때 비상위험준비금은 보험종목별로 경과보험료의 일정비율에 도달할 때까지 기존 적립액, 향후 손해율 추이 등을 고려하여 적립한다. 또한 보험종목별로 경과위험 손해율(발생손해액/경과위험보험료)이 일정비율을 초과하는 경우 그 초과 금액 이내에서 환입할 수 있다. 위와 같은 상황에서 손해액을 정확하게 예측할 수 있다면 회사의 위험부담을 줄여준다. 보험료를 설정할 때부터 손해율이 일정 수준을 넘지 않도록 설정할 수 있게 된다. 회사 입장에서는 갑자기 납입해야 되는 돈을 마련해야 하는 부담이 줄어드는 것이다. 결과적으로 회사의 자산운용에 도움을 줄 뿐만 아니라 경영의 건전성과 효율성 도모라는 긍정적 효과가 있다.

다른 손해보험 종목들과 비교했을 때 자동차보험에는 상대적으로 다양한 일반계약자들이 존재한다. 현대인에게 자동차가 생활필수품이 되면서 자연스럽게 자동차 보험료 또한 가계 지출항목의 필수항목이 되었다. 본 연구를 활용하여 보험금 산출에 관한 정보를 일반보험 계약자가 스스로 알아볼 수 있게 한다면, 정보측면에서 상대적 약자인 계약자를 보호할 수 있다. 자신의 정보를 입력함으로써 받을 수 있는 보험금의 대략적 범위를 파악할 수 있게 되고, 정당한 계약을 맺을 수 있다.

손해액과 관련하여 다수의 일반보험 계약자들과 보험회사 양측에게 모두 긍정적 효과를 불러 일으킬 수 있는 보험종목을 연구했으며 그 결과로 자동차보험을 주제로 채택했다. 모델링, 시각화, 통계적검정 기법을 통해 손해액 산출에 필요한 요인들이 무엇이 있는지 증명하여 논리성을 입증한 후, 실제로 어떤 식으로 적용 가능한 지까지 연구를 진행했다.

II. 본론

1. 해당 연구에 사용된 기법

사용할 자료에는 여러 요인들과 실제 발생 손해액이 변수로 들어가 있다.

설명변수(X)	실제 발생 손해액에 영향을 줄 것으로 예상되는 변수들 Ex) 재소자 여부, 학력차이, 청구한 손해액, 지난 5년간의 손해액의 합, 나이, 성별
종속변수(Y)	실제 지난 5년간 발생한 손해액

어떤 변수들이 주요한 요인으로 작용하는지 모델을 통해 파악한다. 예를 들어 Y= ax + b 라는 x와 y의 관계에서 a가 클수록 x가 조금 변해도 Y가 변하는 크기가 커져서 X가 y에 큰 영향을 주는 요인이 된다. 모델 속에서도 이와 같은 연산이 존재하며 그 과정 속에서 가장 영향력이 큰 변수들을 추리고 관계를 파악해야 한다.

가장 간단한 방법으로 선형회귀가 있다. 종속변수 Y와 한 개 이상의 설명변수 X와의 선형 상관 관계를 모델링하는 기법으로 연속형(숫자형) 자료만 사용 가능하다. 범주형을 다루기 위해서는 연속형으로 임의로 바꿔줘야 하는데 이 과정에서 범주형 변수의 정보가 부정확해질 수 있다. 대게 크기의 차이를 가지고 있어 순서를 나타낼 수 있는 연속형 자료와 달리 범주형 자료는 논리적 순서를 가지거나 가지지 않을 수 있다. 이를 보안하기 위해 범주형 변수와 연속형 변수 모두 이용할 수 있는 결정나무 모델을 본 연구에서 사용한다.

결정나무 모델에서는 질문을 통해 자료를 세분화하며 각각 하위항목에 대한 질문을 바탕으로 자료를 분류한다. 모호성이 적은 질문을 상위에 두고 모호성이 가장 큰 질문을 하위에 둔다. 어떤 질문에 대한 답의 경우의 수가 적은 것이 모호성이 적은 질문이다. 따라서 상위에서 하위로 내려가면서 질문에 대한 대답을 선택하는 과정을 통해 모호성을 줄이게 된다.

다음으로는 변수의 중요도를 파악하기 위해 X축에는 설명변수 Y축에는 종속변수를 두어 시각화 한다. 이를 통해 변수들이 어떠한 방식으로 분포하는지 파악한

다. 위에서 파악한 설명변수와 종속변수 간의 관계를 통계검정 과정에서 증명한다. 반면 변수 간 관계를 파악할 수 없는 것들은 부가적인 통계검정이 필요하다. 위 과정을 통해 증명된 관계들의 논리성(자료의 이질성과 부족 때문에 논리성이 약할 수 있음)을 판단하기 위해 타 연구자료를 참고하고, 통계검정 결과에서 우연적으로 의미를 가지는 관계를 배제한다.

2. 연구에 사용된 자료와 파생변수

본 연구를 위해 마드리드 콤플루텐세 대학교의 논문에서의 자료를 바탕으로 변수를 선별하고 파생변수를 생성한다.[1]

번호	변수명	의미
1	KIDSDRIV	자동차에 태우고 다니는 아이 수
2	BIRTH	태어난 달
3	AGE	나이
4	HOMEKIDS	자녀 명 수
5	YOJ	일에 종사한 기간 (year)
6	INCOME	수입 ($)
7	PARENT1	한 부모 가정 (TRUE: 1 FALSE: 0)
8	HOME_VAL	거주하는 집 가격 ($)
9	MSTATUS	재소자 여부 (TRUE: 1 FALSE: 0)
10	GENDER	성 (MALE: 0 FEMALE: 1)
11	EDUCATION	학력에 따른 가중치 부과 (High School 미만: 0 High School: 1 Bachelors: 2 Masters: 3 PhD: 4)
12	OCCUPATION	직업별 임의로 숫자 지정 (Professional: 1 Blue Collar: 2 Manager: 3 Clerical: 4 Lawyer: 5 Home Maker: 6 Doctor: 7 Student: 8)
13	TRAVTIME	일자리까지의 거리 (minute)
14	CAR_USE	차량 사용 유형 (Private: 1 Commercial: 0)
15	BLUEBOOK	차량 가격 ($)

[1] Maria Manuela Moura e Moura, 「Cálculo de la Prima Pura en un Seguro de Automóvil para la Garantía de Daños Propios, mediante Modelos Lineales Generalizados y Segmentación de Clientes por Conglomerados」, Universidad Complutense de Madrid, 2017, pp.7~8.

16	TIF	보험계약 경과 시간 (year)
17	CAR_TYPE	차량 유형(크기)에 따른 가중치 부과 (Sports Car: 0 SUV: 1 Pickup: 2 Panel Truck: 3 Minivan: 4 Van: 5)
18	OLDCLAIM	지난 5년 간 손해액의 합
19	CLM_FREQ	과거에 운전자가 보험 청구를 한 횟수
20	REVOKED	지난 7년 간 면허취소 여부 (TRUE: 1 FALSE: 0)
21	MVR_PTS	차량 정비검사 횟수
22	CAR_AGE	차량 사용기간 (year)
23	URBANICITY	집이나 직장이 위치한 지역유형 (Highly Rural: 0 Highly Urban: 1)

위의 표로 정리한 변수들 중에서 OLDCLAIM이 종속변수가 되고, 나머지 변수들은 설명변수가 된다. 설명변수들이 지난 5년간 손해액의 합에 어떻게 영향을 끼치는지 연구한다.

3. 연구과정

3-A. 모델을 통한 주요 변수 채택

본 연구에서는 Correlation matrix를 이용하여 다중 공선성을 제거한다. 다중공선성(multicollinearity)이란 독립변수의 일부가 다른 독립변수의 조합으로 표현되는 것을 말한다. 즉, 독립변수 간 상호 상관관계가 강한 경우에 나타난다. 이렇듯이 독립변수가 의존하면 과최적화(over-fitting) 문제가 발생하고, 모델 결과의 안정성을 해치게 된다. 일반적으로 corr 변수의 값이 0.9 이상이면 다중공선성 문제가 발생한다. 이를 방지하기 위해 연구과정에서 의존적인 변수는 삭제하고자 했으나 해당 연구의 자료탐색 결과 다중 공선성이 발견되지 않았다.

```
#corr이 클수록 변수끼리의 연관성이 있는 것이다.
#상관성이 너무 큰 변수들이 많으면 모델의 성능이 떨어진다.
#필요치않은 변수들을 지워 주면 좋다.
#하지만 아래와 같이 max corr가 0.68이므로 지울 변수는 없어 보인다.
def correl(X_train):
    cor = X_train.corr()
    corrm = np.corrcoef(X_train.transpose())
    corr = corrm - np.diagflat(corrm.diagonal())
    print("max corr:",corr.max(), ", min corr: ", corr.min())
    c1 = cor.stack().sort_values(ascending=False).drop_duplicates()
    high_cor = c1[c1.values!=1]
    ## change this value to get more correlation results
    thresh = 0.4
    display(high_cor[high_cor>thresh])
print("다중공선성 파악")
correl(data.drop("OLDCLAIM",axis=1, inplace=False))
```

```
다중공선성 파악
max corr: 0.6850462850184104 , min corr:  -0.6720568995949949

EDUCATION  CAR_AGE     0.685046
INCOME     HOME_VAL    0.592363
           EDUCATION   0.572329
KIDSDRIV   HOMEKIDS    0.458153
MSTATUS    HOME_VAL    0.453913
HOMEKIDS   PARENT1     0.443944
MVR_PTS    CLM_FREQ    0.407438
RED_CAR    CAR_TYPE    0.402684
dtype: float64
```

변수의 중요도를 판단하기 위해 결정나무의 종류 중 하나인 '랜덤 포레스트' 모델을 적용한다. 이것은 데이터를 이용해 다수의 결정 트리를 만들고, 만들어진 결정 트리의 결과들을 모아 다수결로 최종 결과를 도출하는 정확도 높은 알고리즘이다. 정확도를 높이기 위해 Information Gain'과 'Feature Importance'과정을 바탕으로 모델을 생성한다.

Information Gain (정보 이득)[2]: 엔트로피를 이용한 데이터 분할에 사용되는 "정보 이득"도 주목할 만한 용어다. 속성에서 데이터셋을 분할한 후의 엔트로피 감소로 계산한다. 질문 전후의 엔트로피 감소 정도가 클수록 명확한 질문이다.

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

T=종속 변수

X = 분할할 설명 변수 (질문에 이용될 변수)

Entropy(T,X) = 변수 X에서 데이터가 분할된 후 계산된 엔트로피

Feature Importance: 변수 중요도는 해당 노드에 도달할 확률에 의해 가중되는 노드 불순성의 감소로 계산된다. 노드 확률을 노드에 도달하는 샘플 수로 계산하여 총 샘플 수로 나눌 수 있다. 결과값이 높을수록 변수가 중요한 것이다. 아래 과정을 통해서 변수 중요도를 도출할 수 있다.

i. 각 의사결정 트리에서 두 개의 자식 노드(이진수 트리)만 가정하여 지니 중요도를 이용하여 노드 중요성을 계산함.

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

ni sub(j) = node j의 중요성

w sub(j) = 노드 j에 도달하는 샘플의 가중치 수

C sub(j) = 노드 j의 불순물 값

왼쪽(j) = 노드 j의 왼쪽 분할에 따른 자식 노드

right(j)=node j에서 오른쪽 분할된 자식 노드

[2] Stacey Ronaghan, "The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark", 2018.05.11, <https://medium.com/@srnghn/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3>.

ii. 의사결정 트리의 각 형상에 대한 중요도를 다음과 같이 계산함.

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

fi sub(i) = 특징의 중요성 i

ni sub(j)=node j의 중요성

iii. 이 값을 모든 변수의 중요도 합으로 나누어 0과 1 사이의 값으로 정규화

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

iv. 각 나무의 변수 중요도 합을 계산하여 총 나무 수로 나눔.

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$$

RFfi sub(i) = 랜덤 포레스트의 모든 트리에서 계산한 특징의 중요성

normfi sub(ij) = 나무 j에서 i에 대한 정규화된 형상 중요도

T = 총 나무 수

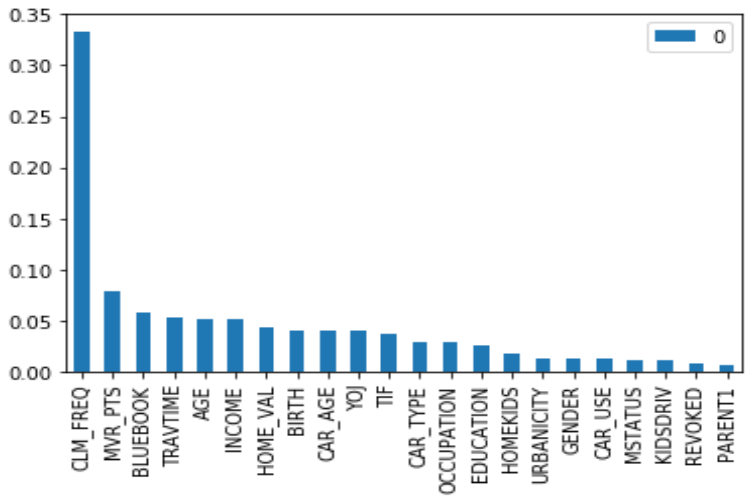
Cross Validation: 모델을 학습시킬 때 주어진 자료를 모두 사용하게 되면 검정용 데이터가 남아있지 않게 되어 모델링이 맞는지 틀린 지 알 수 가 없다. 따라서 전체자료를 n개의 집합으로 나눈 뒤 n-1개의 집합을 모델을 학습시키는 데 사용하고 나머지 하나의 집합은 검정용으로 사용한다. 즉, 검정용 자료로 모든 n개의 집합을 쓰기위해 총 n번의 모델학습이 진행되는 것이다.

```
#Random Forest 모델의 정확성 파악
X=data.drop(columns='OLDCLAIM',axis=1,inplace=False).values
y=data.OLDCLAIM.values
cross_val_score(RandomForestClassifier(random_state=1), X, y, cv = 50).mean()
#정확성                CV
#0.8763035311111139 10
#0.9620018623876502 50
```

본 연구에서 사용한 모델은 집합을 10개로 나눌 시 정확도는 87%, 50개로 나눌 시 96%로 올라갔다. 집합의 개수와 정확도가 정비례하지만 정확도의 변화가 미미한 시점부터 집합의 개수를 고정시킨다. 과도하게 집합을 나눈다면 만든 모델이 주어진 자료에만 적용될 가능성이 있다. 과적합 문제(주어진 자료에만 정확하고 일반적인 자료에는 정확하지 않은 문제)를 해결하기 위해서는 실제 금융 데이터가 다량으로 필요하지만 구할 방도가 없으므로 이를 한계점으로 인정한다.

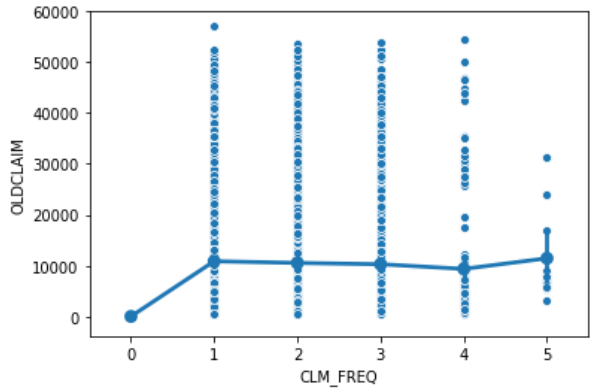
위에서 생성된 모델의 변수 중요도를 그래프로 나타낸다. 도출한 결과를 바탕으로 CLM_FREQ부터 TIF까지의 설명변수들과 OLDCLAIM 간의 관계를 파악한다. 그 외의 설명변수들은 중요도가 너무 낮기 때문에 제외한다.

```
x=data.drop('OLDCLAIM',axis=1)
y=data['OLDCLAIM']
model= RandomForestClassifier(random_state=1)
model.fit(x, y)
print('<설명변수들의 중요도 표와 그래프>')
importances = list(zip(model.feature_importances_, x.columns))
importances.sort(reverse=True)
pd.DataFrame(importances, index=[x for (_,x) in importances]).plot(kind = 'bar')
print(importances)
```



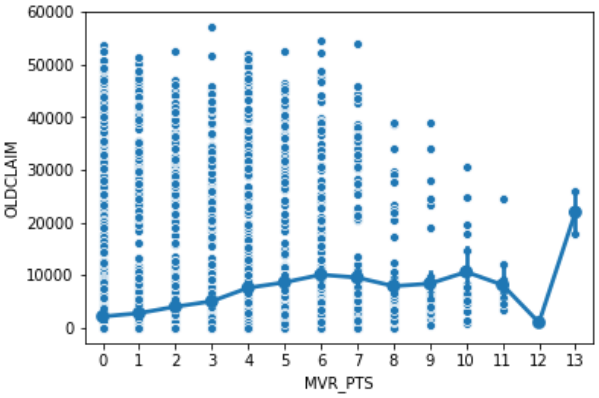
3-B. X축(주요 설명변수)과 y축(종속변수)에 해당하는 자료의 시각화

Scatter plot으로 설명변수의 값에 대응하는 OLDCLAIM 값을 찍고, 설명변수의 각 자료 값에서 다양한 OLDCLAIM 값을 평균 낸 점들을 분포 변화를 표현한다. 자료의 개수가 다 일정하지 않으므로 평균을 내어 특징을 파악한다.



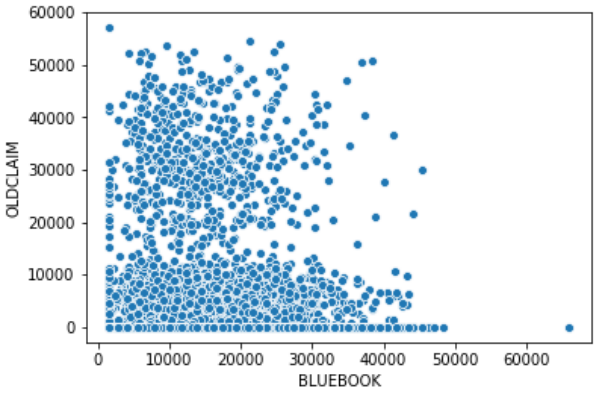
CLM_FREQ

비례 관계가 보인다.
평균점들을 이은 선이
양의 기울기이다.



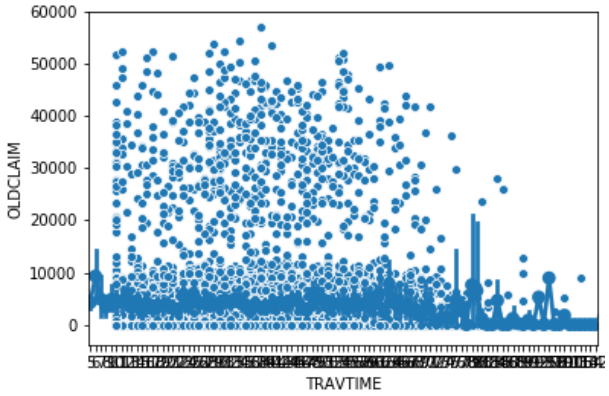
MVR_PTS

종속변수와 비례관계이다. MVR_PTS가 12일 때 하나의 이상치를 제외하면 비례관계라 할 수 있다.



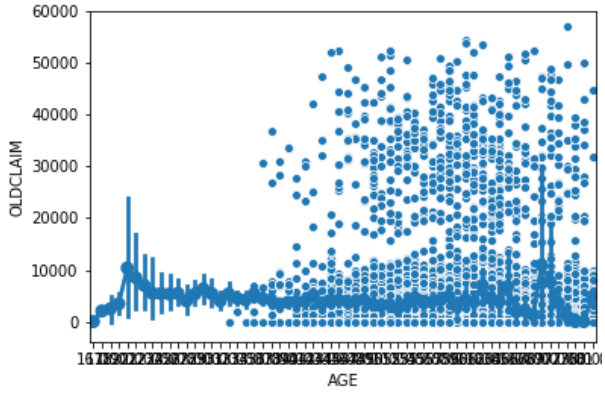
BLUEBOOK

평균 값으로 선을 만들기에 BLUEBOOK의 값이 너무 다양하다. 그저 이 단계에서는 점만 찍어보았다. 검정을 통한 관계 파악이 필요하다.



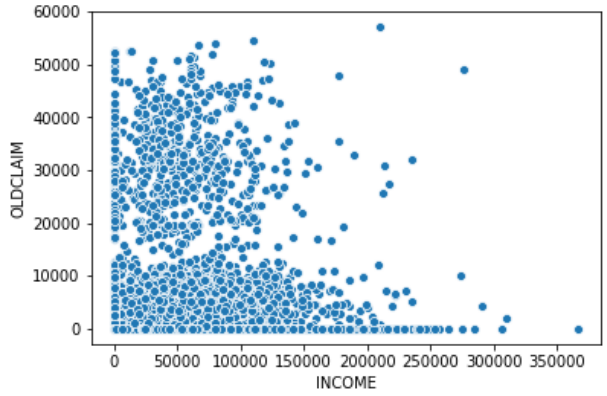
TRAVTIME

기울기가 완만하지만 음수이므로 설명변수가 종속변수와 반비례 관계라고 추측된다.



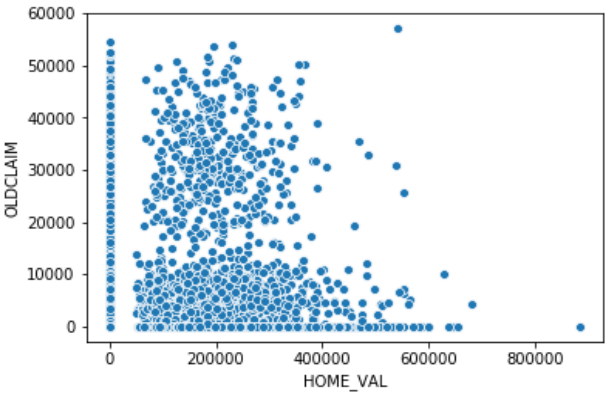
AGE

나이가 적거나 많으면 분포의 패턴이 무너지지만, 중간은 약간의 음의 기울기가 보인다. 반비례 관계라고 추측된다.



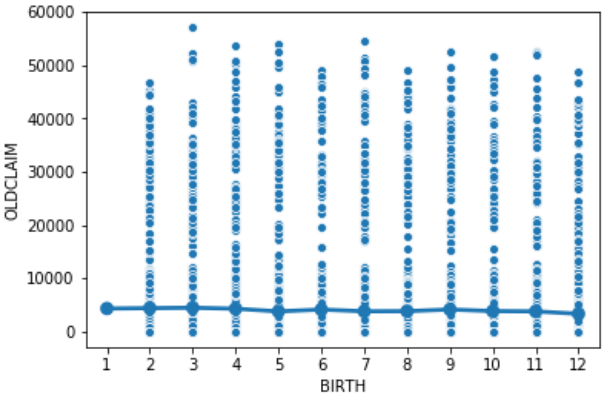
INCOME

평균값으로 선을 만들기에는 INCOME의 값이 다양하므로 이 단계에서는 점만 찍었다. 검정을 통한 관계파악이 필요하다.



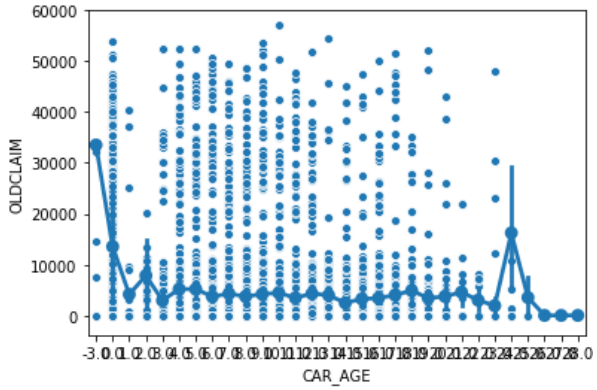
HOME_VAL

평균값으로 선을 만들기에는 HOME_VAL의 값이 다양하므로 이 단계에서는 점만 찍었다. 검정을 통한 관계파악이 필요하다.



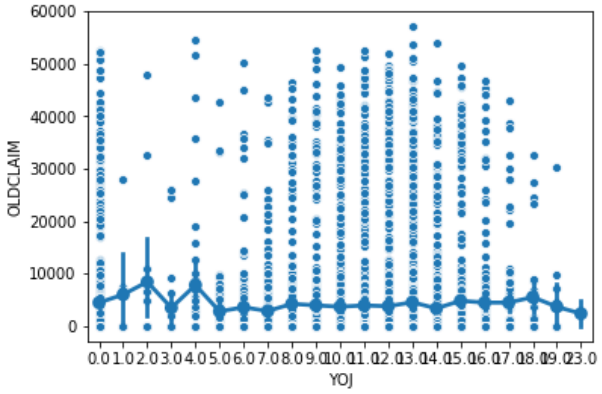
BIRTH

균등 분포에 가깝지만 음의 기울기가 보인다. 반비례 관계가 있다고 예상하지만 정확한 것은 검정이 필요하다.



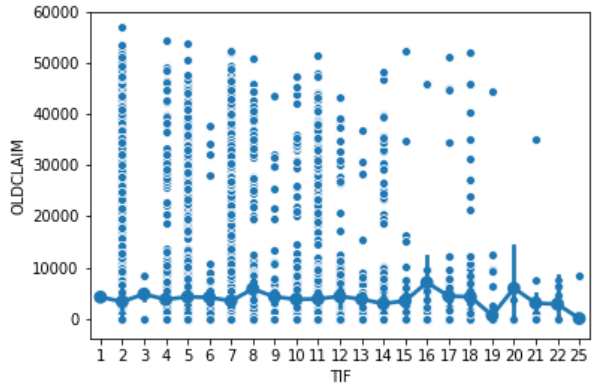
CAR_AGE

자동차가 오래될수록 OLDCLAIM은 낮아지는 것을 볼 수 있다. 설명변수의 값이 25쯤에 이상치가 보이지만 그외에는 반비례 관계이다.



YOJ

분포의 경향성이 자주 바뀌므로 설명변수와 아무 관계가 없을 가능성이 높다.



TIF

분포의 경향성이 자주 바뀌므로 설명변수와 아무 관계가 없을 가능성이 높다.

3-C. 시각화에서 얻은 인사이트를 이용한 통계적 검정

Spearman Rank Correlation Coefficient(순위 상관 계수): 측정형 변수나 순서형 분류형 변수들의 상관관계 정도를 자료의 순위 값에 의하여 계산하는 방법이다. 이 검정의 전제조건은 두 변수에 대한 표본 관측치(X, Y)는 각 변수의 크기순으로 정렬이 가능해야 한다는 것이다.

i. 가설

귀무가설: 두 변수 (X, Y)는 서로 독립이다.

대립가설: 독립이 아니다 = 선형 상관 관계가 존재한다.

ii. 검정 통계량

$$r_R = 1 - \frac{6\Sigma_i {d_i}^2}{n(n^2-1)}$$

n = 데이터 수

di= 고려된 두 변수의 각 i번째 요소의 순위 차

rR의 값으로 t 분포의 p 값을 뽑을 수 있다.

아래 코드를 이용하여 Spearman Correlation Coefficient을 구현한다. 각각의 검정 별로 coefficient(rR)와 p 값이 나오는데, 여기서 p 값은 제 1종 오류(귀무가설이 참인데 기각할 확률)를 뜻한다. 유의 수준을 alpha= 0.05로 지정해주고, p값이 alpha보다 작으면 상관관계성이 있는 것이고 아니면 독립인 것이다. (유의수준은 통계적인 가설검정에서 사용되는 기준 값이다. 일반적으로 유의 수준은 alpha로 표시하고 95%의 신뢰도를 기준으로 한다면 1 - 0.95인 0.05값이 유의수준 값이 된다.) X가 증가할 때 Y가 증가하는 경향이 있다면 Spearman 상관계수는 양수이다. 1에 더 가까울수록 양의 관계가 뚜렷하다. 반면, X가 증가할 때 Y가 감소하는 경향이 있다면 Spearman 상관 계수는 음수이고 -1에 더 가까울수록 음의 관계가 뚜렷하다.

```
coef, p = spearmanr(data.설명변수, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
    print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
    print('Samples are correlated (reject H0) p=%.3f' % p)
```

아래의 표는 위의 과정을 통해 나온 결과를 정리한 것이다. 시각화를 통한 관계파악이 어려운 BLUEBOOK, INCOME, HOME_VAL은 통계적 검정을 통하여 관계성이 입증되었다. 시각화를 통한 관계파악 과정을 거친 변수 중에서는 TRAVTIME과 OLDCLAIM 간의 관계성이 통계적 검정을 통해 반증되고, 나머지 변수들은 관계성이 증명되었다.

변수명	CLM_FREQ	MVR_PTS	BLUE_BOOK	TRAVTIME
Corr Coefficient	0.928	0.431	-0.059	0.002
Correlation	Y	Y	Y	N

변수명	AGE	INCOME	HOME_VAL	BIRTH
Corr Coefficient	-0.052	-0.061	-0.104	-0.027
Correlation	Y	Y	Y	Y

변수명	CAR_AGE	YOJ	TIF
Corr Coefficient	-0.035	-0.012	-0.02
Correlation	Y	N	N

Correlation Y/N: 변수 간 관계가 있다/없다.

관계성이 증명되었지만 자료만의 특성으로 인하여 나온 결과도 있으므로 이는 다음 과정에서 수정하도록 한다.

4. 논리적 관계 도출

본 연구에서는 해외자료를 사용했으며 자료의 개수가 적었으므로 연구결과를 증명하기 위해서는 입증된 다른 근거를 댈 수 있어야 한다. 따라서 상식적으로 생각할 수 있는 논리와 연구에서 근거를 취득한 설명변수를 추리고, 해당 변수들만 실현 가능한 방안의 주 요인으로 채택하기 위해 아래의 과정을 도입한다.

i. CLM_FREQ와 OLDCLAIM간의 양의 상관관계

전체 고객을 대상으로 CLM_FREQ을 계산해보면 대수의 법칙에 의해 정규분포를 따른다. 따라서 평균적으로 보험금을 받을 확률은 n*p로 표현된다(n: 보험금을 청구한 횟수, p: 보험금을 받을 확률). 보험금 청구 횟수인 n이 커지면 n*p도 커져서 지난 5년간 받은 보험금액도 늘어나는 것이다.

ii. MVR_PTS와 OLDCLAIM간의 양의 상관관계

일반적인 지식 수준의 소비자는 자동차 수리와 관련된 의사 결정 과정에 있어서 가격의 적정성과 제품의 품질수준 등에 대하여 객관적 수준을 판단하기 어렵다. 그러므로 사업자와 소비자 간의 정보 취득 범위 및 지식 수준 차이에 따라 상호이해도가 떨어지게 되면 해당 시장의 소비자지향성 역시 낮게 평가될 가능성이 크다. 실제로도 한국의 사례를 보면 2015년도 소비자시장성과지수(CMPI) 분석결과 '자동차수리서비스'는 71.1점으로 가장 낮은 평가를 받았다.[3] 차량 내 문제가 없는 자동차들은 정기 점검을 받지 않는 경향이 있다. 즉, 점검 횟수가 많다는 것은 해당 차량의 안정성 결함과 사고발생 횟수가 많은 것이다. 지난 5년간 받은 보험금액도 자연적으로 늘어나기 때문에 MVR_PTS와 OLDCLAIM 간의 양의 상관관계의 존재는 논리성이 있다.

iii. BLUEBOOK과 OLDCLAIM간의 음의 상관관계

자동차 가격이 높을수록 사고 발생 시 배상금이 크기 때문에 타 차량 운전자가 더 주의를 기울이는 경향이 있다. 즉, 교통사고 발생 확률이 감소하므로 자동차 가격과 지난 5년 간 받은 보험금은 반비례 관계이다.

[3] 정영훈·허민영, 「자동차수리서비스의 시장구조 분석 연구」, 한국소비자원, 2015, p.14.

iv. AGE와 OLDCLAIM간의 음의 상관관계

65세 이상은 26-59세 계층보다 사고 빈도가 낮았으므로[4] 60세를 기준으로 나이가 적은 그룹과 많은 그룹으로 나누었을 시 상관관계가 파악된다. 그러나 연령대 별로 젊은 연령대의 운전자들과 고령 운전자들의 차이를 비교하는 연구가 필요하다.[5]

v. INCOME과 OLDCLAIM간의 음의 상관관계

관련 근거와 연구결과를 찾지 못했지만, 만약 한국 전체의 금융 자료로도 같은 결과가 나온다면 연구해볼 가치가 충분한 상관 관계이다.

vi. HOME_VAL과 OLDCLAIM간의 음의 상관관계

지역별 인적요인이 교통사고율과 상관관계가 매우 높다.[6] 지역마다 다른 집값, 거주지가 사람에게 끼치는 환경적 영향이 큰 점과 지역별 인적요인과 손해액의 정비례 관계를 근거로 집값과 지난 5년간 받은 보험금의 상관관계가 입증된다. 단, 상관 관계의 방향성은 심리학적 연구가 필요하다.

vii. BIRTH와 OLDCLAIM간의 음의 상관관계

태어난 달에 따라 운전자의 성격이 다르다는 논리성의 근거가 없다. 이것은 주어진 자료의 우연적 특성이므로 실현 가능한 방안에서는 제거한다.

viii. CAR_AGE와 OLDCLAIM간의 음의 상관관계

상식적으로 연식이 오래된 자동차가 보험에 가입할 시 연식이 얼마 안 된 자동차보다 손해액이 크므로 보험료가 더 비싸다. 이를 통해 자동차 연식과 손해액 간의 양의 상관 관계가 있음을 확인했다. 본 연구에서는 자료의 부재와 부적합성 때문에 관계의 방향성에서 오류가 나타났다. CAR_AGE를 논리성 결여로 주요변수에서 탈락시켜야 한다.

---

[4] 이경희, 「고연령 자동차 보험 계약자의 사고위험 분석」, 상명대학교, 2015, p.69.

[5] 이미진·이명선, 「고령운전자의 인지된 운전능력과 운전행동 및 사고위험의 관련성」, 한국위기관리논집, 2014, p.302.

[6] 김동국, 「교통사고감소를 위한 자동차보험의 지역요인 반영에 관한 연구」, 교통투자평가협회, 2015, pp.235~236.

5. 실현 가능한 방안

- 보험회사를 위한 방안

위에서 26-59세와 65세 이상으로 운전자 그룹을 나누었으나 세세한 분류가 이루어지지 않았다는 한계점이 있었다. 이 점을 보완하여 보험회사에서 연령대 별로 자세히 그룹화하여 운전자들 간 차이를 연구한다면 나이와 손해액 간 관계를 명확히 하여 손해액 예측에 기여할 것이다.

연봉과 손해액의 반비례 관계에 대하여 전체자료를 이용한 연구가 필요하다. 이 경우에도 반비례 관계를 나타내어 논리성이 입증되면 보험료 또는 손해액 산출 시 연봉 변수를 사용할 수 있다.

손해액에 영향을 주는 지역별 인적요인에 집값이 포함된다는 것을 확인했으나 구체적인 영향 범위는 확인하지 못했다. 집값이 어떤 방향으로 영향을 끼치는 지 입증된다면 보험료 또는 손해액 산출의 주요 변인으로 쓸 수 있다.

전체를 대표할 수 있는 자료에 위의 과정으로 논리성을 취득한 변수를 포함시키고, 랜덤 포레스트나 타 모델을 학습시킨다면, 머신러닝과 딥러닝이 가진 블랙박스 문제(논리적 관계가 없는데 억지로 결과를 끌어내는 것)도 해결된다. 결과적으로 이 모델을 실제 손해액 산출에 쓸 수 있게 된다.

- 일반계약자를 위한 방안

보험회사 측 방안 마지막에서 언급했던 모델링은 실제로 일반계약자에게 더 필요하다. 전문지식이 없는 계약자에게는 손해액 예측 플랫폼이 참고 자료로써 유용하게 쓰일 것이다.

III. 결론

본 연구에서는 일반적인 예측을 위한 모델링이 아닌, 주요 변수를 파악하기 위해 모델링을 사용했다. 연구 과정을 거치면서 여러가지 검정 방법을 통해 손해액과 관계없는 변수를 제거했다. 자료의 부적합성과 부재를 해결하기 위해 일반적인 한국 자료에도 적용될 수 있도록 증명된 관계에 대한 타 연구 결과를 인용하여 논리성을 더했다. 이 과정에서 살아남은 변수들을 손해액 예측 모델에

넣고자 했다. 즉, 회사측에서 본 연구를 제안할 시 불확실한 관계성에 대해 의문이 없어 거절 사유가 없도록 했다. 또한, 논리성 기반 손해액 예측 플랫폼은 일반 계약자들이 자신의 보험료가 합당한 금액인지 판단하는 기준이 될 수 있으므로 사람들이 유용하게 이용할 것이다.

< 참고문헌 >

- 김동국, 「교통사고감소를 위한 자동차보험의 지역요인 반영에 관한 연구」, 교통투자평가협회, 2015.
- 김명준, 「 적정 보험료 수준 예측을 위한 심도 빈도의 추세 분석에 관한 연구」, 한남대학교, 2013.
- 이경희, 「고연령 자동차 보험 계약자의 사고위험 분석」, 상명대학교, 2015.
- 이미진·이명선, 「고령운전자의 인지된 운전능력과 운전행동 및 사고위험의 관련성」, 한국위기관리논집, 2014.
- 이현수, 「 신뢰도 적용방법에 따른 자동차보험 가격 산출」, 중앙대학교, 2010.
- 정영훈·허민영, 「 자동차수리서비스의 시장구조 분석 연구」, 한국소비자원, 2015.
- Kaggle, “Car Insurance Claim Data”, https://www.kaggle.com/xiaomengsun/car-insurance-claim-data, 2019.02.09
- Maria Manuela Moura e Moura, 「Cálculo de la Prima Pura en un Seguro de Automóvil para la Garantía de Daños Propios, mediante Modelos Lineales Generalizados y Segmentación de Clientes por Conglomerados」, Universidad Complutense de Madrid, 2017.
- Medium, “The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark”, https://medium.com/@srnghn/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3, 2019.02.24.

< 코드 리스트 >

```
#사용한 Python 패키지들
import pandas as pd
import numpy as np
from numpy import mean
import re
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
import seaborn as sb
from scipy.stats import spearmanr
```

```
pd.set_option('display.max_columns', None)
#쓸 자료를 가져옴.
data=pd.read_csv("car_insurance_claim.csv")

#종간에 어느 설명변수라도 값이 없다면, 모델에 쓸 수 없으므로 그냥 다 삭제해줌.
#결측값에 mean, median 값등을 넣어서 모델을만들어도 되지만, 우리가 중점에 두는 것은 정확성이기 때문에
#자료를 삭제 하기로 결정.
#결측값을 지운 후 자료의 개수 : 7857
data.dropna(how="any", inplace=True)


del data["CLAIM_FLAG"]
#사고가 난다면 지출될 손해액의 예측값이 므로 필요가 없다.
del data["CLM_AMT"]
print(data.head())
del data["RED_CAR"]

#ID변수는 손해액에 영향을주는변수가 아니므로 삭제
del data["ID"]

#생년월일에서 월만 뽑아내서 새로운 x변수를 만듦. (태어난 월이 손해액에 영향을 미칠 수 도 있다고 판단함.)
data.BIRTH =data.BIRTH.str[2:5]
data.BIRTH.replace({'JAN':1, 'FEB':2, 'MAR':3, 'APR':4, 'MAY':5, 'JUN':6, 'JUL':7, 'AUG':8, 'SEP':9,
                    'OCT':10, 'NOV':11, 'DEC':12}, inplace=True)

#INCOME에서 $와 ,를 없애고 int로 바꿔줌.
data.INCOME = data.INCOME.str.strip('$')
data.INCOME=pd.to_numeric(data.INCOME.str.replace(",",""))

#PARENT1에서 No면 0으로 Yes면 1로 바꿔줌.
data.PARENT1.replace({'No':0, 'Yes':1},inplace=True)

#HOME_VAL에서 $와 ,를 없애고 int로 바꿔줌.
data.HOME_VAL= data.HOME_VAL.str.strip('$')
data.HOME_VAL=pd.to_numeric(data.HOME_VAL.str.replace(",",""))

#MSTATUS에서 z_No면 0으로 Yes면 1로 바꿔줌.
data.MSTATUS.replace({'z_No':0, 'Yes':1},inplace=True)

#GENDER에서 M면 0으로 z_F면 1로 바꿔줌.
data.GENDER.replace({'M':0, 'z_F':1},inplace=True)

#CAR_USE에서 Commercial면 0으로 Private면 1로 바꿔줌.
data.CAR_USE.replace({'Commercial':0, 'Private':1},inplace=True)

#BLUEBOOK에서 $와 ,를 없애고 int로 바꿔줌.
data.BLUEBOOK= data.BLUEBOOK.str.strip('$')
data.BLUEBOOK=pd.to_numeric(data.BLUEBOOK.str.replace(",",""))


#OLDCLAIM에서 $와 ,를 없애고 int로 바꿔줌.
data.OLDCLAIM= data.OLDCLAIM.str.strip('$')
data.OLDCLAIM=pd.to_numeric(data.OLDCLAIM.str.replace(",",""))

#REVOKED에서 No면 0으로 Yes면 1로 바꿔줌.
data.REVOKED.replace({'No':0, 'Yes':1},inplace=True)
```

```
#URBANICITY에서 z_Highly Rural/ Rural면 0으로 Highly Urban/ Urban면 1로 바꿔줌.
data.URBANICITY.replace({'z_Highly Rural/ Rural':0, 'Highly Urban/ Urban':1},inplace=True)

#EDUCATION부한 정도에 따라 가중치를 줌
data.EDUCATION.replace({'<High School':0, 'z_High School':1,'Bachelors':2,
                        'Masters':3, 'PhD':4},inplace=True)
#CAR_TYPE 자동차 크기에 따라서 가중치를 줌.
data.CAR_TYPE.replace({'Sports Car':0, 'z_SUV':1, 'Pickup':2, 'Panel Truck':3, 'Minivan':4, 'Van':5}, inplace=True)

#OCCUPATION 그냥 아무 숫자나 arbitrary 하게 지정해줌. 값들 사이에 임의로 차이를 부여하기 애매하다.
data.OCCUPATION.replace({'Professional':1, 'z_Blue Collar':2, 'Manager':3, 'Clerical':4, 'Lawyer':5, 'Home Maker':6,
                         'Doctor':7, 'Student':8}, inplace=True)
```

```
#corr이 클수록 변수끼리의 연관성이 있는 것이다.
#상관성이 너무 큰 변수들이 많으면 모델의 성능이 떨어진다.
#필요치않은 변수들을 지워 주면 좋다.
#하지만 아래와 같이 max corr가 0.68이므로 지울 변수는 없어 보인다.
def correl(X_train):
    cor = X_train.corr()
    corrm = np.corrcoef(X_train.transpose())
    corr = corrm - np.diagflat(corrm.diagonal())
    print("max corr:",corr.max(), ", min corr: ", corr.min())
    c1 = cor.stack().sort_values(ascending=False).drop_duplicates()
    high_cor = c1[c1.values!=1]
    ## change this value to get more correlation results
    thresh = 0.4
    display(high_cor[high_cor>thresh])
print("다중공선성 파악")
correl(data.drop("OLDCLAIM",axis=1, inplace=False))


x=data.drop('OLDCLAIM',axis=1)
y=data['OLDCLAIM']
model= RandomForestClassifier(random_state=1)
model.fit(x, y)
print('<설명변수들의 중요도 표와 그래프>')
importances = list(zip(model.feature_importances_, x.columns))
importances.sort(reverse=True)
pd.DataFrame(importances, index=[x for (_,x) in importances]).plot(kind = 'bar')
print(importances)
```

```
#Random Forest 모델의 정확성 파악
X=data.drop(columns='OLDCLAIM',axis=1,inplace=False).values
y=data.OLDCLAIM.values
cross_val_score(RandomForestClassifier(random_state=1), X, y, cv = 50).mean()
#정확성              CV
#0.8763035311111139 10
#0.9620018623876502 50
```

```
sb.scatterplot('CLM_FREQ', 'OLDCLAIM', data = data)
sb.pointplot('CLM_FREQ', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.CLM_FREQ, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: 0.928
Samples are correlated (reject H0) p=0.000
```

```
sb.scatterplot('MVR_PTS', 'OLDCLAIM', data = data)
sb.pointplot('MVR_PTS', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.MVR_PTS, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: 0.431
Samples are correlated (reject H0) p=0.000
```

```
sb.scatterplot('BLUEBOOK', 'OLDCLAIM', data = data)
coef, p = spearmanr(data.BLUEBOOK, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.059
Samples are correlated (reject H0) p=0.000
```

```
sb.scatterplot('TRAVTIME', 'OLDCLAIM', data = data)
sb.pointplot('TRAVTIME', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.TRAVTIME, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: 0.002
Samples are uncorrelated (fail to reject H0) p=0.886
```

```
sb.scatterplot('AGE', 'OLDCLAIM', data = data)
sb.pointplot('AGE', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.AGE, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.052
Samples are correlated (reject H0) p=0.000
```

```
sb.scatterplot('INCOME', 'OLDCLAIM', data = data)
coef, p = spearmanr(data.INCOME, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.061
Samples are correlated (reject H0) p=0.000
```

```
sb.scatterplot('HOME_VAL', 'OLDCLAIM', data = data)
coef, p = spearmanr(data.HOME_VAL, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.104
Samples are correlated (reject H0) p=0.000
```

```
sb.scatterplot('BIRTH', 'OLDCLAIM', data = data)
sb.pointplot('BIRTH', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.BIRTH, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.027
Samples are correlated (reject H0) p=0.017
```

```
sb.scatterplot('CAR_AGE', 'OLDCLAIM', data = data)
sb.pointplot('CAR_AGE', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.CAR_AGE, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.035
Samples are correlated (reject H0) p=0.002
```

```
sb.scatterplot('YOJ', 'OLDCLAIM',data=data)
sb.pointplot('YOJ', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.YOJ, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.012
Samples are uncorrelated (fail to reject H0) p=0.298
```

```
sb.scatterplot('TIF', 'OLDCLAIM', data = data)
sb.pointplot('TIF', 'OLDCLAIM', data=data, estimator= mean)
coef, p = spearmanr(data.TIF, data.OLDCLAIM)
print('Spearmans correlation coefficient: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
        print('Samples are uncorrelated (fail to reject H0) p=%.3f' % p)
else:
        print('Samples are correlated (reject H0) p=%.3f' % p)
```

```
Spearmans correlation coefficient: -0.020
Samples are uncorrelated (fail to reject H0) p=0.075
```