

운전자 정보 및 자동차 정보에 따른 자동차 사고 보험 청구 예측

<데이터 마이닝 팀 프로젝트>



20182471 김지민
20201362 조인영

CONTENTS

1. 연구 목적
2. DATA
3. DATA Preprocessing
4. Modeling
5. 결론
6. 한계점 및 기대방향

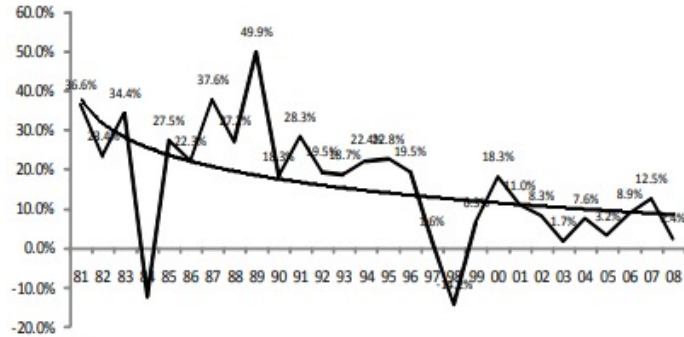


이

연구 목적

연구 목적

〈그림 II-1〉 자동차보험시장 수입보험료 연도별 성장률 추이



자료 : 기승도(2010), p.20.

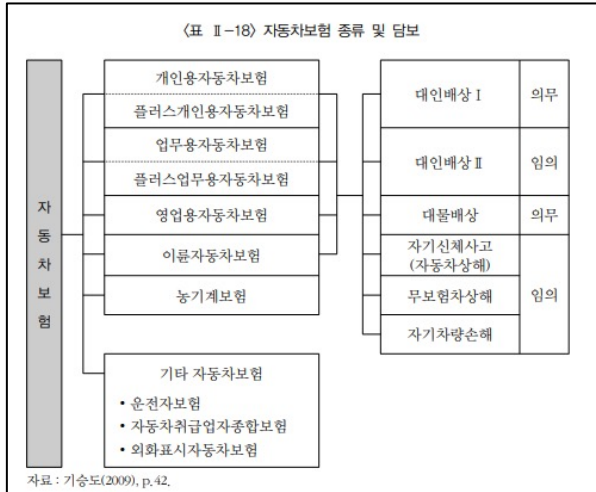
〈표 II-1〉 연대별 수입보험료 연평균 성장률

구 분	1980년대	1990년대	2000년대
연평균성장률	26.2%	13.1%	6.8%

자료 : 기승도(2010), p.20-21.

- 현 자동차보험시장의 규모는 감소하고 있는 추세이다.
- 이는 자동차보험 산업의 경쟁환경이 더욱 심해지고 있다는 것을 의미한다.
- 보험 회사에서는 기존 고객의 이탈에 대해 더욱 신경 쓰고 고객 만족을 높일 수 있는 방법을 찾아야 한다.

연구 목적



- 기업은 기존 고객의 이탈방지를 위한 기존 고객 맞춤형 상품을 창출해 내야한다.
- 자동차 보험 회사는 운전자와 자동차의 많은 정보들 중 어떤 것이 보험 청구에 가장 영향을 미치는지 파악 하고 보험 청구 여부를 정확하게 예측할 필요가 있다.
- 머신 러닝과 딥 러닝을 통해 분석해 청구 여부에 대해 예측하고 해당 정보를 통해 고객에게 가장 알맞은 보험 종류를 제공, 상품 경쟁 전략에서 앞서 나갈 수 있도록 한다

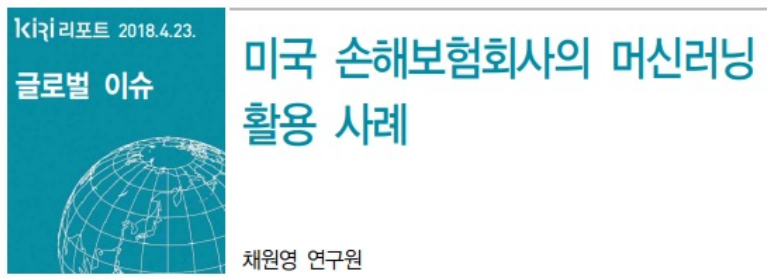
유사 연구 현 상황

KB손보, '머신러닝' 활용 '자동차보험 AI자동심사 시스템' 개발

▲ 홍윤기 기자 | © 입력 2022.11.23 18:21 | 댓글 0



KB손해보험 '자동차보험 AI자동심사 시스템' 개발./KB손해보험



- 많은 보험 회사들도 보험 청구 예측, 사고 발생 데이터를 통한 실시간 수리비 산정에 머신 러닝을 도입하고 있음



02

DATA

DATA

- Kaggle에서 제공하는 Car Insurance Claim Prediction Data를 사용.
- 총 43개의 자동차와 운전자에 대한 정보로 구성되어 있다. (1개의 target)
- 범주형 데이터와 수치형 데이터로 이루어져 있다.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 58592 entries, 0 to 58591  
Data columns (total 44 columns):
```

0	policy_id	58592	non-null	object	24	steering_type	58592	non-null	object
1	policy_tenure	58592	non-null	float64	25	turning_radius	58592	non-null	float64
2	age_of_car	58592	non-null	float64	26	length	58592	non-null	int64
3	age_of_policyholder	58592	non-null	float64	27	width	58592	non-null	int64
4	area_cluster	58592	non-null	object	28	height	58592	non-null	int64
5	population_density	58592	non-null	int64	29	gross_weight	58592	non-null	int64
6	make	58592	non-null	int64	30	is_front_fog_lights	58592	non-null	object
7	segment	58592	non-null	object	31	is_rear_window_wiper	58592	non-null	object
8	model	58592	non-null	object	32	is_rear_window_washer	58592	non-null	object
9	fuel_type	58592	non-null	object	33	is_rear_window_defogger	58592	non-null	object
10	max_torque	58592	non-null	object	34	is_brake_assist	58592	non-null	object
11	max_power	58592	non-null	object	35	is_power_door_locks	58592	non-null	object
12	engine_type	58592	non-null	object	36	is_central_locking	58592	non-null	object
13	airbags	58592	non-null	int64	37	is_power_steering	58592	non-null	object
14	is_esc	58592	non-null	object	38	is_driver_seat_height_adjustable	58592	non-null	object
15	is_adjustable_steering	58592	non-null	object	39	is_day_night_rear_view_mirror	58592	non-null	object
16	is_tpm	58592	non-null	object	40	is_ecw	58592	non-null	object
17	is_parking_sensors	58592	non-null	object	41	is_speed_alert	58592	non-null	object
18	is_parking_camera	58592	non-null	object	42	ncap_rating	58592	non-null	int64
19	rear_brakes_type	58592	non-null	int64	43	is_claim	58592	non-null	int64
20	displacement	58592	non-null	int64					
21	cylinder	58592	non-null	int64					
22	transmission_type	58592	non-null	object					
23	gear_box	58592	non-null	int64					

DATA



POWER STEERING

자동차의 핸들 조작을 쉽게 하기 위한 자동차의 장치 일종.



ESC

[전자 제어 주행 안전 장치]

차량의 안정성을 향상시키고 미끄러짐을 예방하기 위한 능동안전기술을 말한다.



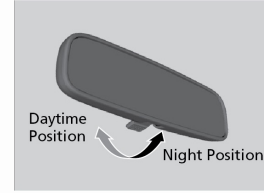
TURNING RADIUS

차량의 회전 직경
차량이 회전하는데 필요한 최소 직경을 말한다.



DISPLACEMENT

엔진 배기량



DAY NIGHT REAR VIEW MIRROR

주야간 거울.
뒷 차의 헤드라이트 눈부심 방지 기능이 포함된 거울



GROSS WEIGHT

차량 총 중량



03

DATA PREPROCESSING

Data Preprocessing

- Classification 모델 사용을 위해 범주형 변수들을 Label encoding을 통해 수치형 변수로 바꾸어 주었다.
- One-hot encoding을 사용할 경우 attribute 개수가 너무 많아 짐을 고려해 Label encoding 방법을 택하였다.

< Label encoding 실행 코드 >

```
for columns in tqdm(df.columns):  
    if dict(df.dtypes)[columns] == 'object':  
        label_encoder = preprocessing.LabelEncoder()  
        df[columns] = label_encoder.fit_transform(df[columns])
```

< Label encoding 실행 후 data 모습 >

is_central_locking	is_power_steering	is_driver_seat_height_adjustable	is_day_night_rear_view_mirror	is_ecw	is_speed_alert	ncap_rating
0	1	0	0	0	1	0
0	1	0	0	0	1	0
0	1	0	0	0	1	0
1	1	1	1	1	1	2
1	1	0	1	1	1	2

Data Preprocessing – 변수 선택

※ attribute의 개수가 너무 많을 경우 변수 간의 상관관계가 높아져 모델의 성능이 저하될 염려가 있다.

주성분 분석 (PCA)

- 널리 사용되는 축소 기법 중 하나
- 원데이터의 분포를 최대한 보존하면서 고차원 공간의 데이터들을 저 차원 공간으로 변환하는 기법
- 기존의 변수를 조합하여 서로 연관성이 없는 새로운 변수, 주성분을 만들어 내는 기법

K-Best selection

- scikit-learn 에서 제공하는 모듈
- target변수와 attribute간의 상관관계를 계산하여 K개의 가장 높은 상관관계를 가지는 변수를 선택해주는 모듈
- F-Regression, Chi2(카이 제곱방식) 방법 존재.
- 회귀문제에서 F-regression, 분류 문제에서 chi2방식이 사용.

주성분 분석 (PCA)

< PCA 10개의 기여율 표 >

	설명가능한 분산 비율(고윳값)	기여율	누적기여율
pca1	19.256794	0.458487	0.458487
pca2	8.241030	0.196212	0.654699
pca3	2.539814	0.060471	0.715170
pca4	2.270465	0.054058	0.769227
pca5	1.541541	0.036703	0.805930
pca6	1.323057	0.031501	0.837431
pca7	1.161504	0.027654	0.865085
pca8	1.125472	0.026796	0.891882
pca9	1.058706	0.025207	0.917089
pca10	0.828582	0.019728	0.936817



- 10개의 주성분을 새로 만들어 낸 이후이 고유 값과 기여율의 표
- 제 1 주성분의 기여율이 0.46정도 밖에 되지 않음을 확인할 수 있다.
- 제 3주성분부터는 기여율이 매우 작아져 큰 의미를 찾기 어렵다.

K-Best Selection

< 20개의 선택된 변수 >

```
Selected names: Index(['policy_id', 'policy_tenure', 'age_of_car', 'area_cluster',  
    'population_density', 'segment', 'model', 'fuel_type', 'max_torque',  
    'max_power', 'is_adjustable_steering', 'displacement', 'steering_type',  
    'length', 'width', 'gross_weight', 'is_front_fog_lights',  
    'is_brake_assist', 'is_driver_seat_height_adjustable',  
    'is_day_night_rear_view_mirror'],  
    dtype='object')
```

Alerts

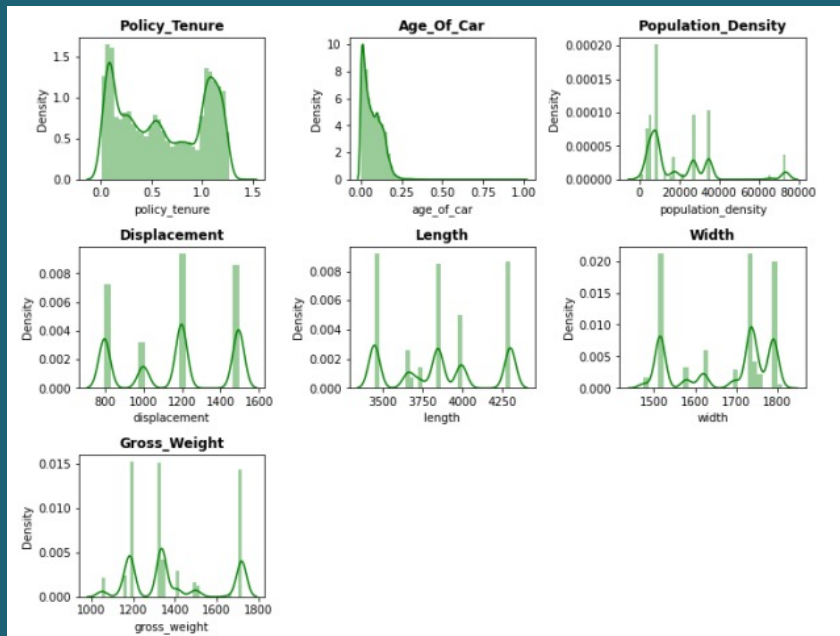
<code>segment</code> is highly correlated with <code>model</code> and 13 other fields	High correlation
<code>model</code> is highly correlated with <code>segment</code> and 13 other fields	High correlation
<code>max_torque</code> is highly correlated with <code>segment</code> and 13 other fields	High correlation
<code>max_power</code> is highly correlated with <code>segment</code> and 13 other fields	High correlation
<code>displacement</code> is highly correlated with <code>segment</code> and 11 other fields	High correlation
<code>length</code> is highly correlated with <code>segment</code> and 13 other fields	High correlation
<code>width</code> is highly correlated with <code>segment</code> and 13 other fields	High correlation
<code>gross_weight</code> is highly correlated with <code>segment</code> and 13 other fields	High correlation
<code>fuel_type</code> is highly correlated with <code>segment</code> and 13 other fields	High correlation
<code>is_adjustable_steering</code> is highly correlated with <code>segment</code> and 12 other fields	High correlation
<code>steering_type</code> is highly correlated with <code>segment</code> and 9 other fields	High correlation
<code>is_front_fog_lights</code> is highly correlated with <code>segment</code> and 12 other fields	High correlation
<code>is_brake_assist</code> is highly correlated with <code>segment</code> and 11 other fields	High correlation
<code>is_driver_seat_height_adjustable</code> is highly correlated with <code>segment</code> and 12 other fields	High correlation
<code>is_day_night_rear_view_mirror</code> is highly correlated with <code>segment</code> and 12 other fields	High correlation
<code>area_cluster</code> is highly correlated with <code>population_density</code>	High correlation
<code>population_density</code> is highly correlated with <code>area_cluster</code>	High correlation

- 20개의 보험 청구 여부와 가장 상관관계가 높은 변수 선택
- 변수 축소 이후에도 변수간 상관관계가 아직 높음을 확인할 수 있다.
- 하지만 너무 많은 변수 축소는 모델 예측의 의미를 희석시킬 수 있기 때문에 더 이상 축소하지 않기로 함.

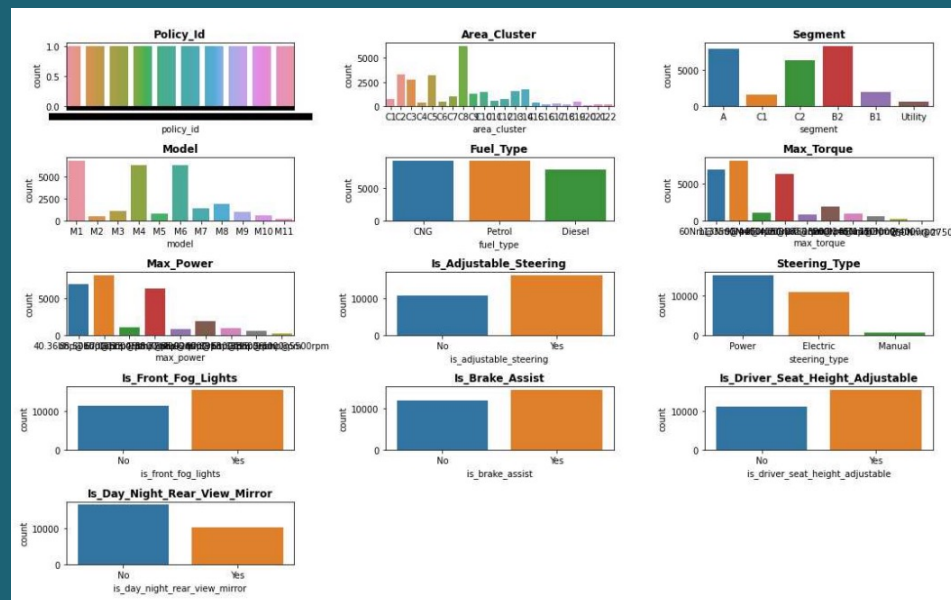
➔ K-Best Selection 방법을 사용하기로 결정

EDA FOR UNDERSTANDING DATA

<수치형 데이터 분포>

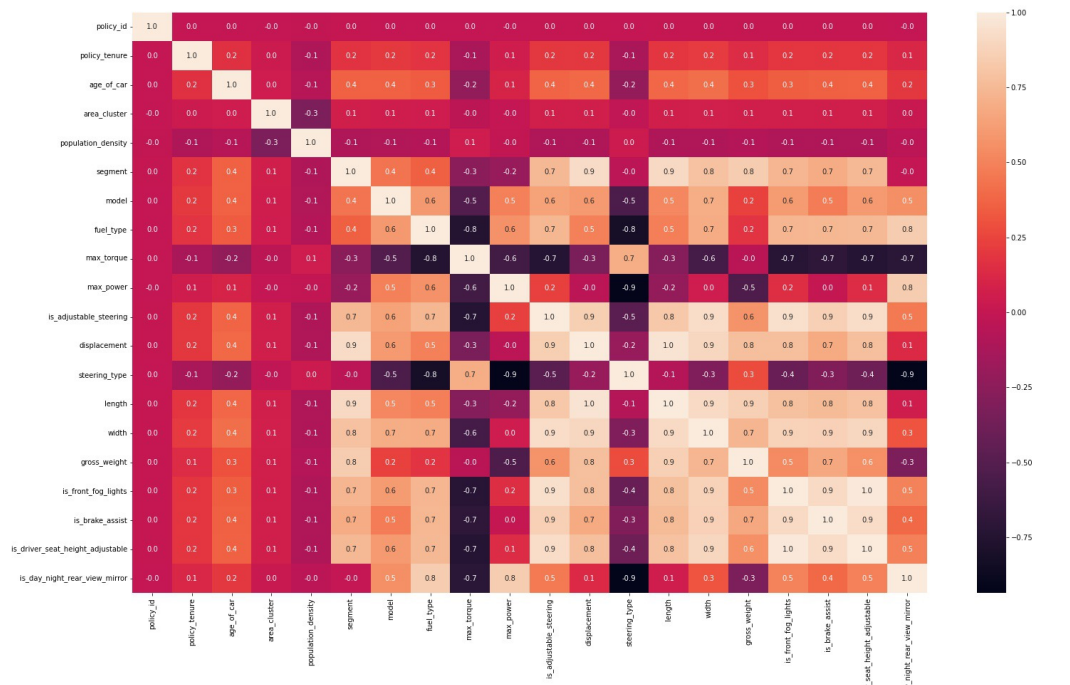


<범주형 데이터 분포>



EDA FOR UNDERSTANDING DATA

< 상관계수 Matrix >



TARGET

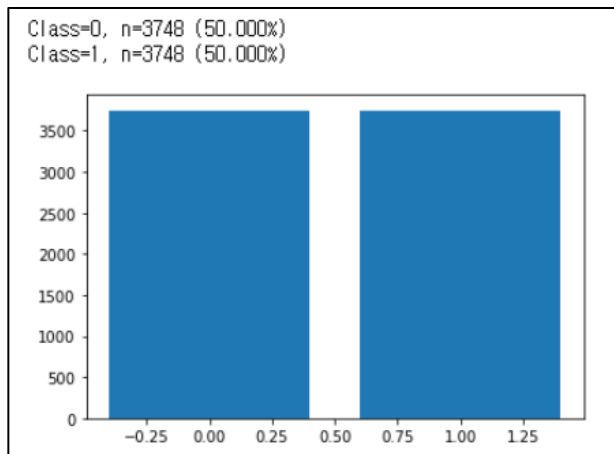


- 자동차 보험 청구라는 Target 특성상 데이터 불균형이 불가피하다.
- 하지만 모두 class 0로 예측하면 accuracy는 높아지는 상황이 발생

→ Target값의 Balance가 요구됨.

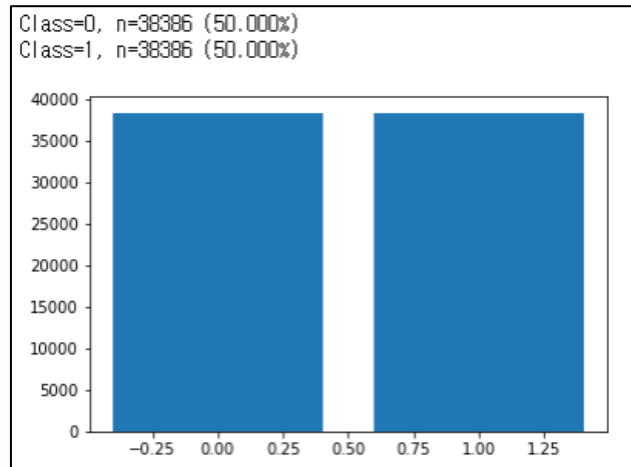
RESAMPLING

< Undersampling >



- 더 낮은 Target의 개수에 데이터의 개수를 맞추는 방법

< Oversampling -Smote >



- 낮은 수의 데이터를 유사한 가상의 데이터를 만들어 균형을 맞춰주는 방법

➔ Undersampling 방법을 사용하기로 결정

PREPROCESSING

< Train-Test split >

```
from sklearn.feature_selection import f_regression, chi2, SelectKBest
from sklearn.model_selection import train_test_split #train, test set 분리
y = df['is_claim']
X = X
train_x, test_x, train_y, test_y = train_test_split(X, y, test_size=0.3, random_state=0)
count_and_plot(train_y)
count_and_plot(test_y)
```

< Min-Max scale>

#데이터 스케일

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler #표준화시키기 위한 패키지
scaler=MinMaxScaler()
scaler.fit(train_x_over)
df_scaled=scaler.transform(train_x_over) #연속형 변수 표준화
df_scaled=pd.DataFrame(data=df_scaled, columns=train_x.columns) #표준화된 데이터를 data frame으로 바꿔주기
```

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

04

MODELING

MODELING

<Classification 평가 척도 종류>

1. Accuracy
2. Precision
3. Recall
4. F1-score
5. ROC Curve
6. AUC



자동차 보험 청구 데이터는 Negative 비율이 매우 높음 (Positive 발생 가능성 희박)

→ 희박한 가능성으로 발생할 상황에 대한 분류지표인 f1-score 로 평가척도 결정

- Precision: $(TP)/(TP+FP)$
- Recall: $(TP)/(TP+FN)$
- F1 score: $(2 \times \text{precision} \times \text{recall})/(\text{precision} + \text{recall})$

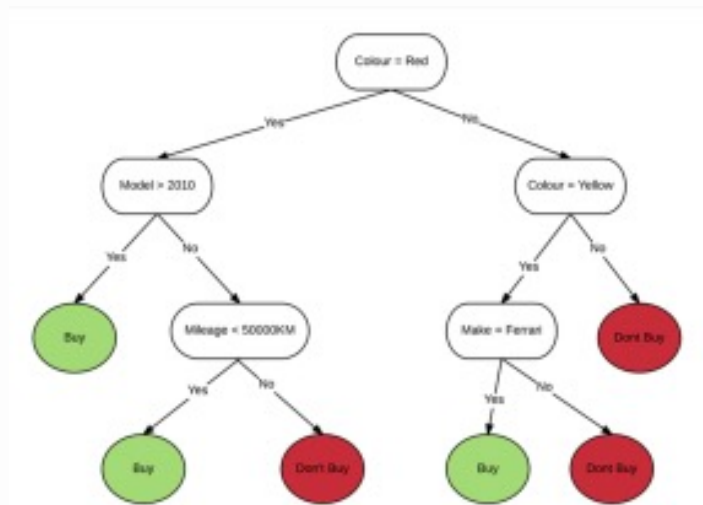
MODELING

활용 알고리즘	
Neural Network	Decision Tree
Gradient Boosting	Logistic Regression
XGBoost	K-Nearest-Neighbor
ADA Boosting	Random Forest

➔ Decision Tree의 F1 score 성능이 가장 우수

DECISION TREE

- Supervised Learning
- 의사 결정 트리
- 데이터의 특성에서 간단한 결정 규칙을 정해 값을 예측
- 나무 시각화 가능
- 트리가 너무 복잡해지면 과적합이 일어날 가능성이 커짐



<데이터 규칙을 만들기 위해 필요한 속성>

- 결정노드
- 리프노드
- 서브트리

MODELING EVALUATION

사용된 전처리 방식

변수선택법	데이터 Unbalance 처리
주성분 분석	Undersampling
Select-K-Best	OverSampling(Smote)
Select-K-Best	Undersampling

→ 3가지의 전처리 방식을 8가지 모델에 적용시킴

MODELING EVALUATION

```
The test f1_score of Neural Network is 0.0  
The test f1_score of XGBoost is 0.5022466300549177  
The test f1_score of Decision Tree is 0.6461318051575932  
The test f1_score of Logistic Regression is 0.018965517241379307  
The test f1_score of KNN is 0.5468025949953661  
The test f1_score of Gradient Boosting is 0.5493406093678945  
The test f1_score of Random Forest is 0.36321031048623315  
The test f1_score of Ada Boosting is 0.5493406093678945  
[Select-K-Best,undersampling]
```

➔ Select-K-Best, Undersampling을 이용한 Decision Tree모델의 성능이 가장 좋았음

OPTIMAL HYPERPARAMETER

<Decision Tree의 최적 hyperparameter선정 >

```
From sklearn.model_selection import GridSearchCV
```

```
Params={'max_depth': [2,3,4,6,8], 'min_samples_split' : [2,3], 'min_samples_leaf' : [4,6]}
```

```
Gs=GridSearchCV(DecisionTreeClassifier(criterion='entropy'), params, scoring='f1', cv=5, verbose=1)
```

- max_depth : 트리의 최대깊이
- Min_samples_split : 리프노드가 되기 위해 필요한 최소한의 샘플 데이터 개수
- Min_samples_leaf : 노드를 분할하기 위한 최소한의 샘플 데이터 개수
- CV: 교차검증 분할 개수

MODELING EVALUATION

<Decision Tree의 최적의 hyperparameter>

GridSearchCV 최고 평균 정확도 수치: 0.6761

GridSearchCV 최적 하이퍼파라미터: {'max_depth': 2, 'min_samples_leaf': 4, 'min_samples_split': 2}

<Decision Tree의 최적의 hyperparameter>

dt_precision: 0.5446853516657852

dt_recall: 0.9074889867841409

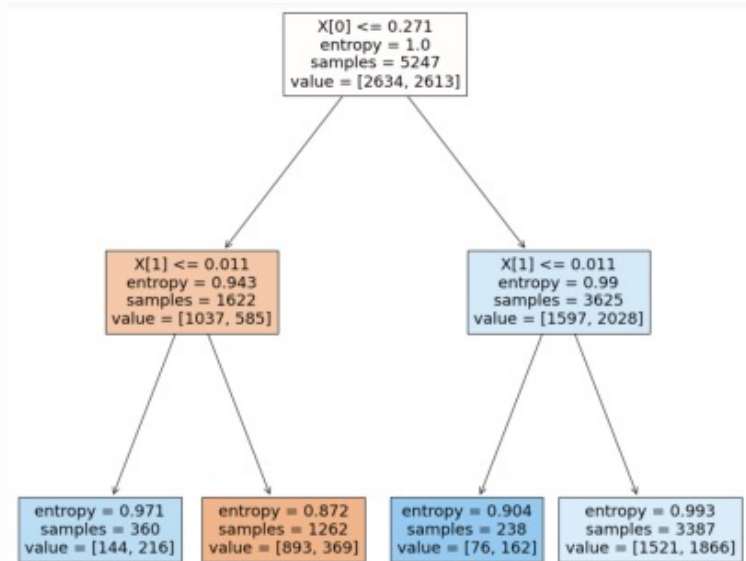
dt_f1-score: 0.6807666886979511

» F1 score 0.64 → 0.68로 향상

```
[ 253  861]
[ 105 1030]
```

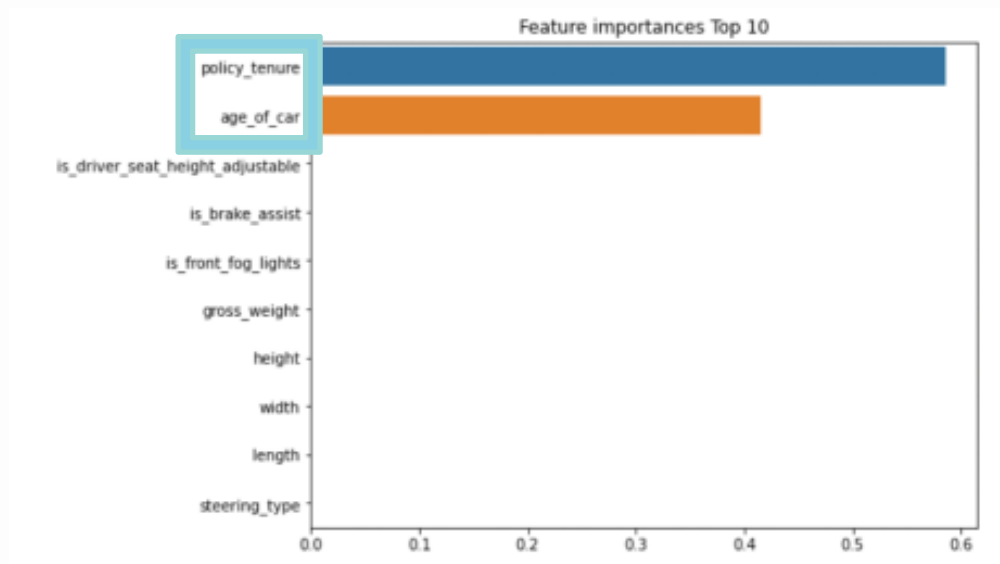
Confusion matrix

VISUALIZATION OF DECISION TREE



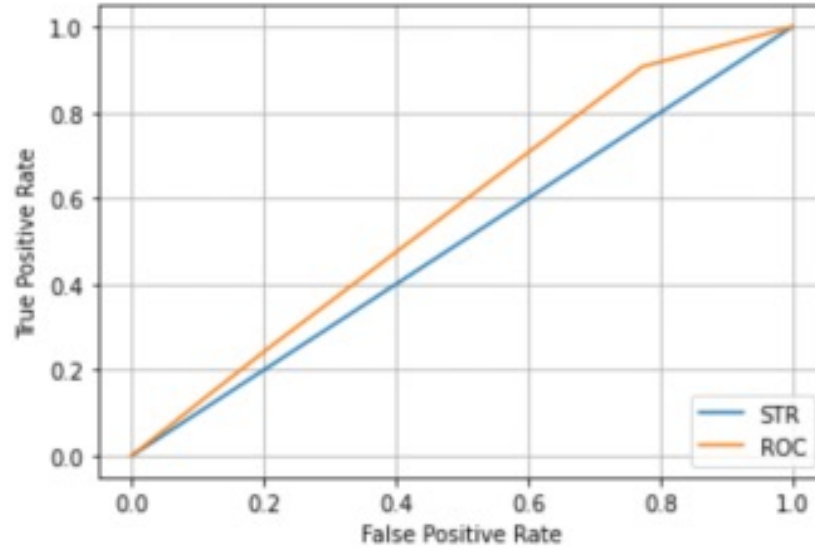
[max_depth=2, min_samples_leaf=4, min_samples_split=2 인 decision tree]

VARIABLE IMPORTANCE



➔ 가장 중요한 변수: Policy_tenure, age_of_car

ROC CURVE




➔ Roc Curve 값이 0.5와 가까운 값을 보여
다소 아쉬움이 있었음

05

CONCLUSION


LIMITATIONS

<데이터 불균형>


- 데이터의 불균형문제를 undersampling 으로 해결
 - Undersampling 을 진행하는 과정에서 자동차 보험 청구를 예측하는데 중요한 데이터가 소실되었을 가능성이 있음
 - 불균형문제를 해결하기 위해 다양한 방식을 시도해볼 필요성이 있음(adasyn, GAN)
- 

LIMITATIONS

<변수 선택>

- Plicy_tenure, age of car를 제외하고 변수의 중요도가 0에 가까움
 - Decision Tree을 적용할 때 중요했을 변수가 전처리의 변수선택 과정에서 빠졌을 가능성이 있음
 - 자동차 보험 가입 여부와 관련된 논문을 통해 연관관계가 높은 변수들을 직접 뽑아내는 방식도 고려해야함
- 

향후 발전 방향

- 최근에 나오는 classification에 유용한 알고리즘을 바탕으로 다양한 모델을 사용해볼 필요가 있음
 - 모든 모델에 대해 최적의 parameter값을 찾거나 세부적인 속성 값들을 변경해 나갈 필요가 있었음
 - 본 프로젝트에 사용된 데이터의 변수 종류에 보험과 관련된 운전자의 정보보단 자동차의 기능적인 것에 대한 정보가 대부분이었음
 - 자동차 보험 청구를 예측하는데 더 도움이 되는 변수들을 추가할 필요가 있음
- 

참고문헌

- 기승도, 황진태 . (2011.03). 충성도를 고려한 자동차보험 마케팅 전략 연구 보험 연구원 . 정책 보고서 권호[: 11-3]
- 포인트 데일리 . “KB손보, 머신러닝 활용 ‘자동차보험 AI 자동심사 시스템’ 개발”. 2022.11.23.
<https://www.thekpm.com/news/articleView.html?idxno=139206>
- Kiri report 제 433호 미국 손해보험회사의 머신 러닝 활용 사례.
- Kaggle. (2022). Car Insurance Claim Prediction.
<https://www.kaggle.com/datasets/ifteshanajnin/carinsuranceclaimprediction-classification/code?select=test.csv>