Stroke Prediction Dataset

11 clinical features for predicting stroke events

Week2 : Preprocessing

Data Mining – 이원상 교수님
실습 2주차

김서연 김연재 윤수진 이가람

Content

1. Intro

현업에서 사용하는 데이터에는 결측치를 갖거나 사용하기 어려운 형태를 갖는 데이터들이 포함되어 있다. 이러한 데이터들은 분석 시 데이터의 품질을 떨어뜨리고 분석의 효용을 낮춘다. 따라서 데이터 분석과 모델 학습에 사용되는 데이터의 형태로 데이터를 가공하는 전처리 과정은 반드시 수반되어야 한다. 본 실습에서는 우리 조가 선택한 주제에 대해 research question을 제시하고, 이를 답하는데 필요한 데이터 분석 과정에 적절한 전처리 과정을 적용하려고 한다.

세계보건기구(WHO)에 따르면 뇌졸중은 전 세계 사망 원인 2위인 치명적인 질병이다. 급작스럽게 찾아오는 응급질환인 만큼 발병 시기를 예측하기도 어렵다. 그 중 고혈압이 있거나, 당뇨병 환자이거나, 흡연을 한다면 그 위험이 더욱 커진다. 실제로 하루 40개비 이상의 담배를 피우는 사람은 10개비 이하의 담배를 피우는 사람에 비해 뇌졸중에 걸릴 가능성이 2배 더 높다. 이는 흡연이 기저질환이 있는 환자의 뇌혈관 손상을 가속화시키고 뇌경색의 가능성을 더욱 높이는 요인이 되기 때문이다. 그렇다면 하루 40개비 이상의 담배를 피우는 흡연자가 10개비 이하의 담배를 피우는 비흡연자에 비해 뇌졸중 진단 가능성이 높다는 가설은 타당한가? 심장병등의 기저질환이 있는 환자의 경우 뇌졸중에 걸릴 가능성이 더 높은가? 흡연 여부 이외에 뇌졸중에 영향을 미치는 요소가 존재하는가?

환자들의 데이터를 분석해 뇌졸중의 발병 여부를 예측한다면 뇌졸중이 찾아올 공포를 줄이는 것은 물론, 발병하지 않도록 더욱 철저히 예방까지 할 수 있을 것이다. 본 실습 과정은 이 데이터 세트의 성별, 나이, 각종 질병, 흡연 상태와 같은 입력 매개변수를 전처리하고, 각 데이터의 분포를 확인한다. 나아가 최종적으로 각 요소가 뇌졸중에 미치는 영향을 확인한다.

2. Research Question

2.1. 흡연자의 경우, 비흡연자에 비해 뇌졸중에 걸릴 가능성이 상대적으로 높은가?

2.2. 기저질환이 있을 경우 뇌졸중에 걸릴 가능성이 높은가?

2.3. 성별, 나이, 각종 질병 여부, 거주 지역이나 결혼 여부 등의 생활양식을 통해 뇌졸중의 예측이 가능한가?

3. Dataset

3.1 Data Explanation

Research Question에 답하기 위하여 'Kaggle'에서 제공하는 'Stroke Prediction Dataset'[1]을 사용하였다. 해당 데이터셋은 IEEE 저널에 실린 논문, 'Identifying Stroke Indicators Using Rough Sets'[2]에 의하면 McKinsey & Company가 관리하는 Electronic Health Record(EHR)을 기반으로 두고 있다. 분석에 활용한 이 데이터셋은 5110x12의 2차원 tabular data의 형태를 띄고 있다. 각각의 속성은 stroke에 영향을 끼칠만한 요소들로 구성되어 있는데, 자세한 속성 정보는 다음과 같다.

<Attribute Information>

1) id: 환자 분별 id
2) gender: 성별, "Male", "Female", "Other"
3) age: 환자의 나이
4) hypertension: 환자에게 고혈압이 없는 경우 0, 있는 경우 1
5) heart_disease: 환자에게 심장병이 없는 경우 0, 있는 경우 1
6) ever_married: 결혼 여부, "No", "Yes"
7) work_type: 고용 상태, "children", "Govt_jov", "Never_worked", "Private", "Self-employed"
8) Residence_type: 거주 지역, "Rural", "Urban"
9) avg_glucose_level: 평균 혈중 글루코스(당) 수치
10) bmi: 체질량 지수
11) smoking_status: 흡연 상태, "formerly smoked", "never smoked", "smokes", "Unknown"*
12) stroke: 환자에게 뇌졸중이 있었을 경우 1, 아닌경우 0

*Note: smoking_status 컬럼에서 환자의 상태를 알 수 없을 경우 "Unknown"으로 표기하였다.

```
[ ] df.head()
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	Female	61	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	Female	79	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

각 속성의 실제 데이터 타입을 확인해보았을 때, gender, ever_married, work_type, Residence_type, smoking_status는 object(string), 나머지의 경우 float과 int등 수치형 데이터가 혼합되어있었다.

[1] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

[2] Pathan, Muhammad Salman & Zhang, Jianbiao & John, Deepu & Nag, Avishek & Dev, Soumyabrata. (2020). Identifying Stroke Indicators Using Rough Sets. IEEE Access. 8. 10.1109/ACCESS.2020.3039439.

하지만 의미적으로 수치형이 아닌 범주형으로 분류 되어야하는 hypertension, heart_disease, stroke는 모두 string타입으로 변경해주었고, age 컬럼의 경우 1세 이하의 영유아에 대해서만 소수점을 가지고 나머지의 경우 int형식을 가지고있었기 때문에, 전반적으로 소수점을 버림하여 int로 통일하였다.

```
[ ]  df.dtypes

id                    int64
gender               object
age                 float64
hypertension          int64
heart_disease         int64
ever_married         object
work_type            object
Residence_type       object
avg_glucose_level   float64
bmi                 float64
smoking_status       object
stroke                int64
dtype: object
```

```
[ ]  df['age'] = df['age'].apply(lambda x: int(x))
     df.drop('id', axis=1, inplace=True)
     for i in ['gender','hypertension', 'heart_disease', 'ever_married',
               'work_type', 'Residence_type', 'smoking_status', 'stroke']:
       df[i] = df[i].apply(lambda x: str(x))
```

3.2 EDA

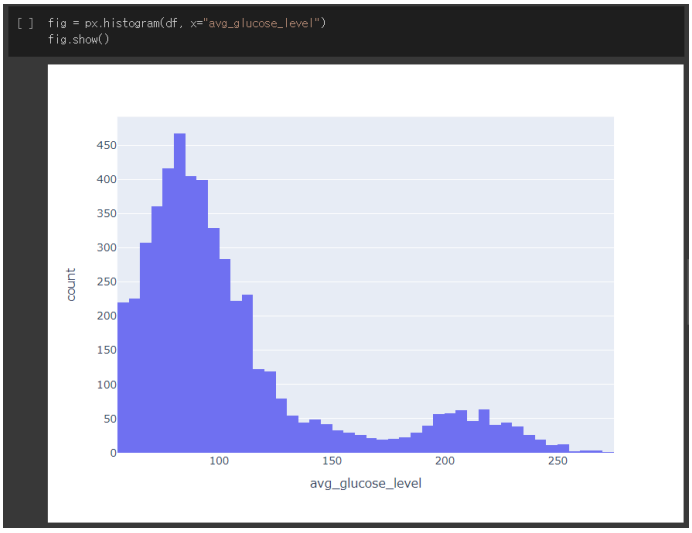
앞서 제시한 Research Question과 관련하여, 각 변수의 분포확인이나 stroke와 다른 변수간의 상관관계 확인, 추가 전처리 아이디어를 얻기 위한 EDA를 진행하였다. EDA를 위한 시각화 라이브러리로는 seaborn, matplotlib 그리고 plotly를 활용하였다.

3.2.1 수치형 변수

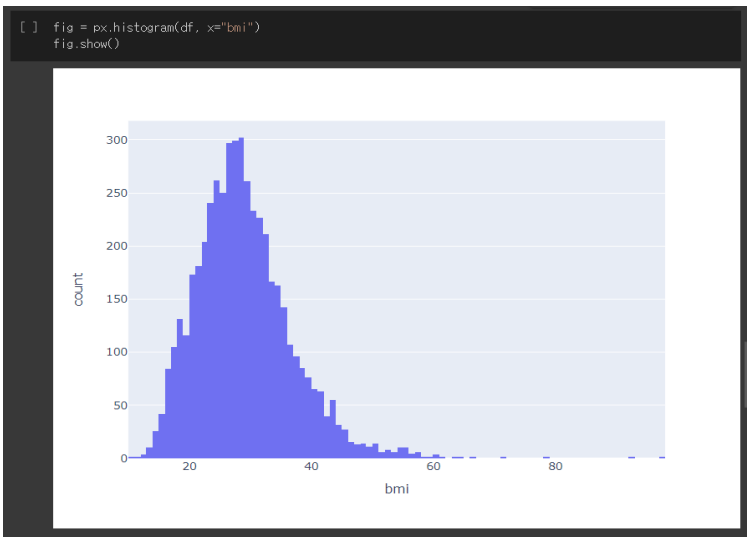
수치형 변수에 대해서는 히스토그램으로 그 분포를 확인해 보았다.

A. avg_glucose_level의 히스토그램 : 정규분포 봉우리가 두개가 보이는 Gaussian Mixture Model로 보거나 왼쪽으로 치우친 분포라고 볼 수 있다. 특별히 이상치라고 볼만한 수치는 육안으로는 찾기 어려웠다.

```
[ ]  fig = px.histogram(df, x="avg_glucose_level")
     fig.show()
```



B. bmi의 히스토그램 : 정규분포와 매우 유사하게 보인다. 하지만 이상치로 보이는 레코드가 몇 건 있을 것으로 예상되어 box plot을 통한 확인을 거친 후 결과에 따라 이상치에 민감한 min-max scaling은 지양하는 것이 올바른 방향으로 보인다.

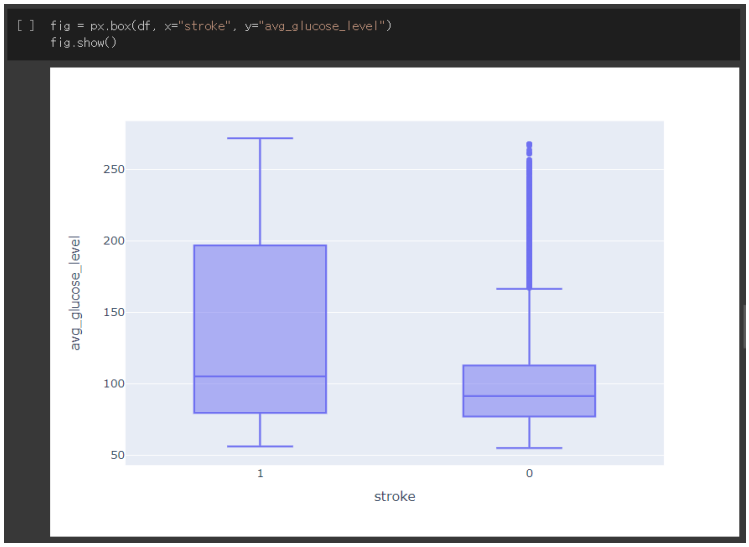


C. age의 히스토그램 : 1세 이하의 영유아에게서 0.8과 같은 float형 데이터가 있음을 확인하였다. 이를 모두 소수점 첫째자리 버림하여 age를 interger로 통일한 후 히스토그램을 확인해보았다. 다른 수치형 변수에 비해 비교적 고른 분포를 보이는 것을 확인할 수 있었다. 특별한 이상치 또한 확인할 수 없었다.

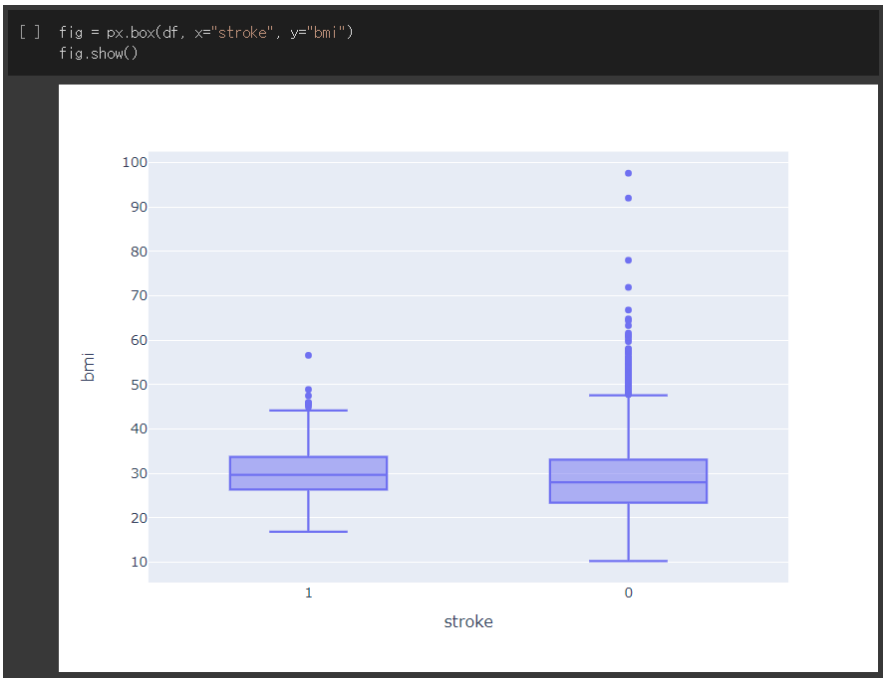
3.2.2 수치형 변수의 stroke과의 상관관계

다음으로는 예측해야할 label인 stroke를 기준으로 각 수치형 분포를 boxplot으로 시각화해보았다
.

A. avg_glucose_level : stroke여부에 따른 avg_glucose_level의 분포 보았을때 두 분포의 중앙값과 q3가 크게 다른 것을 확인할 수 있었다. 이에 avg_glucose_level은 stroke 예측/분류에 좋은 변수라고 판단할 수 있다.

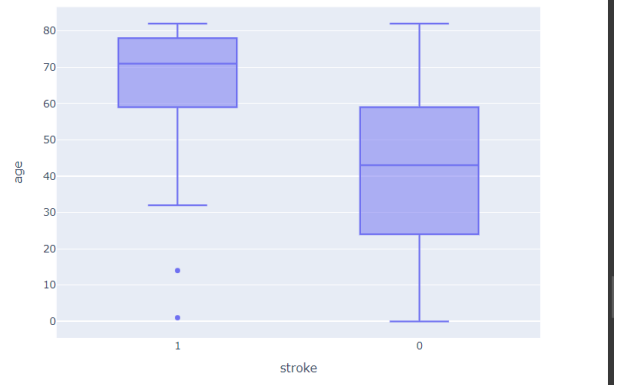


B. bmi : stroke여부에 따른 bmi의 분포 보았을때 0의 경우 표본의 수가 많아 넓게 퍼져있는 반면 1의 경우 비교적 좁게 분포하였다. 중앙값 및 q1,q2,q3가 크게 다르지 않아 bmi의 stroke과의 상관관계를 추가로 확인해야할 필요가 있다.



C. age : 반면 나이의 경우 stroke와 큰 연관성이 있다고 볼 수 있었다. stroke 0에서는 나이가 전반적으로 고르게 분포하는 반면, 1에서는 0~30세 사이에 분포하는 건수는 2건에 불과했고 나머지는 모두 30대 이상의 나이를 가지는 것을 확인하였다.

```
[ ]  fig = px.box(df, x="stroke", y="age")
     fig.show()
```



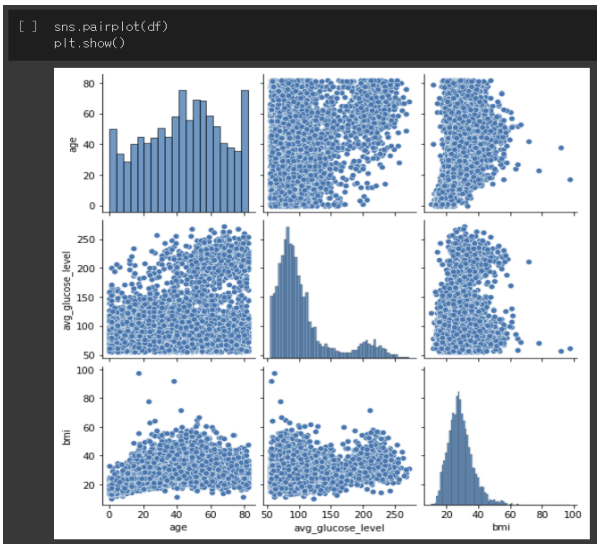
추가로 수치형 변수 간 pearson 상관계수도 확인해보았다. 절댓값 0.3~0.4이상을 상관관계가 크다고 이야기할 수 있는데, 눈에띄게 큰 상관관계를 가지는 수치형 변수는 확인하기 어려웠다.

```
[ ]  df.corr()
```

	age	avg_glucose_level	bmi
age	1.000000	0.238060	0.333738
avg_glucose_level	0.238060	1.000000	0.175502
bmi	0.333738	0.175502	1.000000

마지막으로 모든 수치형 변수의 각 분포와 변수간 산점도를 다음과 같이 한번에 확인할 수 있다.

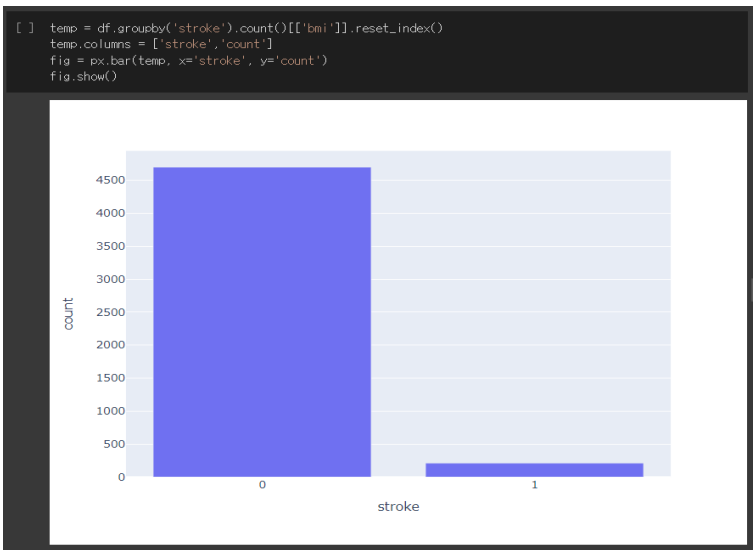
```
[ ]  sns.pairplot(df)
     plt.show()
```



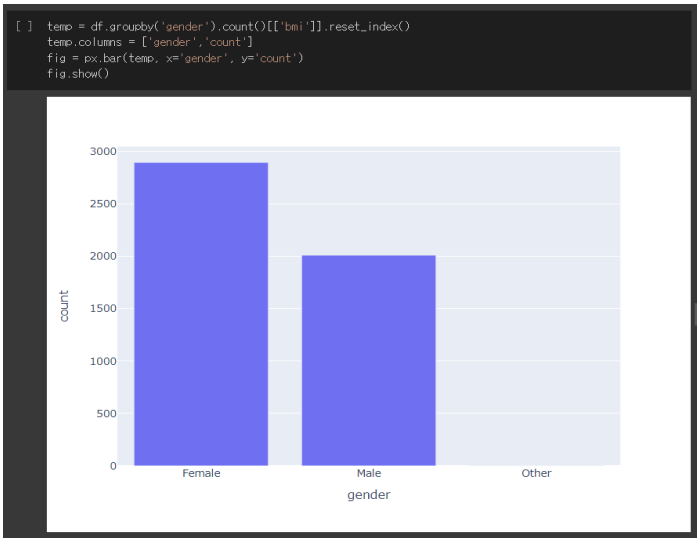
3.2.3 범주형 변수

A. Target인 stroke의 label 분포 : 0인 레코드 수는 4861, 1인 레코드 수는 249로 1인 레코드가 전체의 4.9%인 것을 확인할 수 있었다. 우리의 research question 중 stroke의 예측/분류 문제를 풀기 위해서는 이러한 imbalanced dataset을 그대로 활용할 경우 모델이 데이터를 제대로 학습하지 못하는 문제가 발생할 수 있다. 이러한 문제를 방지하기 위해 Ra

ndom oversampling이나 범주형 데이터와 수치형데이터 모두에 적용 가능한 SMOTE-NC를 활용한 전처리가 필요할 것으로 보인다.



B. gender의 분포 : 전체 5110건 중 2994건이 여성, 2115건이 남성, 1건이 other인 것을 확인하였다. other의 경우 전체에서의 비율이 매우 적고 의미적으로 모호하므로 추후 해당 레코드를 제거할 필요가 있다.



C. 나머지 범주형 데이터에 대해서는 한번에 각 라벨의 수를 barplot으로 시각화해보았다. 그 결과 고혈압 여부, 심장병 발생 여부에 대해서는 0,1의 비율이 크게 다른것을 확인할 수 있었다. 그 외의 경우 대부분의 라벨이 비교적 고르게 분포하였다. 각 범주형 데이터에 대해 라벨의 종류가 2가지인 경우 0,1로 인코딩할 수 있다.

그와 달리 smoking_status의 라벨은 never smoked, Unknown, formerly smoked, smokes로 총 4가지 이다. 나쁨, 중간, 좋음과 같은 단계적 범주형데이터의 경우 1,2,3과 같이 증가하는 숫자로 인코딩할 수 있겠지만, smoking status의 경우 Unkown이 포함되어 있어

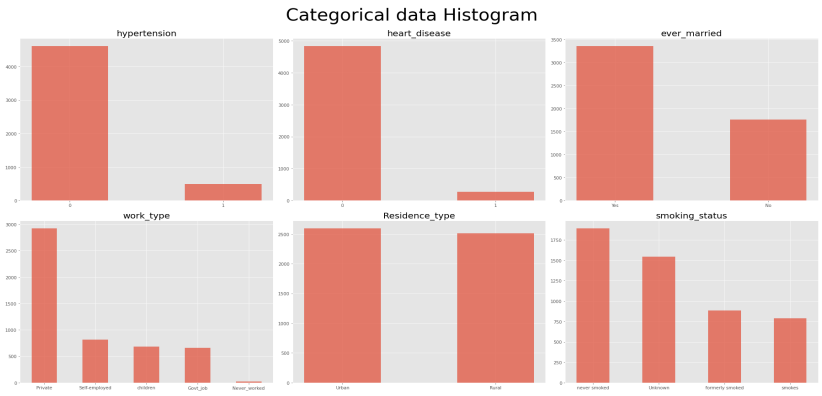
위의 예시처럼 인코딩하기는 어렵다고 판단하여 차후 categorical encoding시 이를 고려하여 진행하였다.

work type의 경우 라벨이 여러개인 점, 각 라벨 간의 관계를 대,소 상,하 관계로 보기 어려운 점을 고려하여 one-hot encoding을 진행하였다.

```
[ ]  plt.style.use("ggplot")

     # 히스토그램 을 사용해서 데이터의 분포를 살펴봅니다.
     plt.figure(figsize=(25,20))
     plt.suptitle("Categorical data Histogram", fontsize=40)

     # id는 제외하고 시각화합니다.
     cols = ['hypertension', 'heart_disease', 'ever_married', 'work_type',
             'Residence_type','smoking_status']
     for i in range(0, len(cols)):
         plt.subplot(3,3,i+1)
         plt.title(cols[i], fontsize=20)
         temp = df[cols[i]].value_counts()
         plt.bar(temp.keys(), temp.values, width=0.5, alpha=0.7)
         plt.xticks(temp.keys())
     plt.tight_layout(rect=[0, 0.03, 1, 0.95])
     plt.show()
```



3.2.4 범주형 변수의 stroke과의 상관관계

앞서 제시한 3가지 Research question중 나이, 평균 혈중 당수치, bmi와 stroke의 관련성을 파악하고자 2.3.2에서 boxplot을 통해 stroke의 여부에 따른 수치형 변수의 분포를 확인하였다. 성별, 질병 여부, 흡연 여부와 같은 범주형 변수와 stroke의 상관관계를 확인하기 위해서는 앞서 활용한 방법으로는 그 상관관계를 확인하기 어렵다. 이러한 범주형 데이터간의 상관관계는 Cramers'V 계수로 확인해볼 수 있다.

$$V = \sqrt{\frac{\chi^2}{n\,(q-1)}}$$

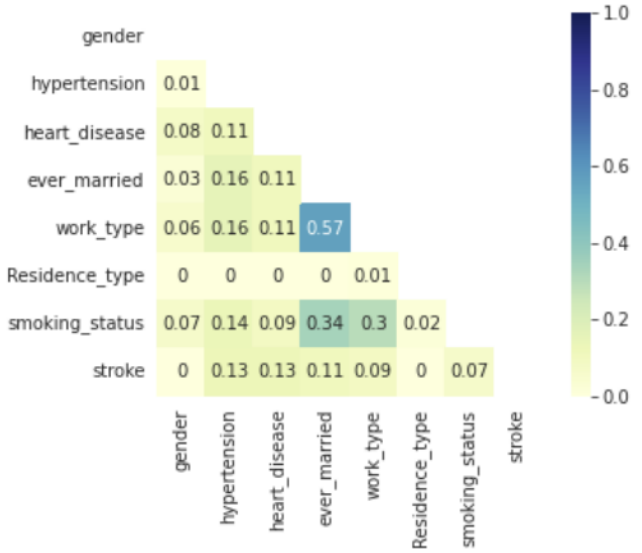
χ^2 : 카이자승 공식에 의해 구함

n : 총사례수

q : 줄 또는 칸의 유목수 중에 적은 숫자

```
[ ]  def cramers_V(var1,var2) :
       crosstab = np.array(pd.crosstab(var1, var2, rownames=None, colnames=None)) # Crosstab building
       chi2 = chi2_contingency(crosstab)[0]
       n = np.sum(crosstab)
       phi2 = chi2 / n
       r, k = crosstab.shape
       phi2corr = max(0, phi2 - ((k-1)*(r-1))/(n-1))
       rcorr = r - ((r-1)**2)/(n-1)
       kcorr = k - ((k-1)**2)/(n-1)
       return np.sqrt(phi2corr / min((kcorr-1), (rcorr-1)))
```

위와 같이 계산할 수 있는 Cramers'V 계수는 두개의 범주형 변수가 얼마나 강력하게 연관되는지를 표현한다.이 값은 항상 0~1사이의 양수값을 가지는데, 1에 가까울수록 그 연관관계가 크다고 해석할 수 있다. 대체로 계수 ≤ 0.2일때는 약한 상관관계,0.2 < 계수 ≤ 0.6는 적당한 상관관계, 계수 > 0.6의 경우 강한 상관관계를 가진다고 해석한다.



결과는 위의 히트맵과 같이 나타난다. work_type과 ever_married 컬럼의 상관관계, smoking status와 ever_married, work_type 외에는 모두 두 범주형 변수 간 약한 상관관계를 나타낸다. 하지만 이는 단순 두 개 변수만의 상관관계로, 여러개 변수가 합쳐진 상관관계 효과는 확인할 수 없다. 그렇기 때문에 해당 변수들을 현재 단계에서 모두 제거하기 보다는, 이후 전진선택법이나 후진소거법과 같은 변수 선택 단계를 거쳐 변수를 제거하거나 삭제하도록 한다.

3.3 Imputation

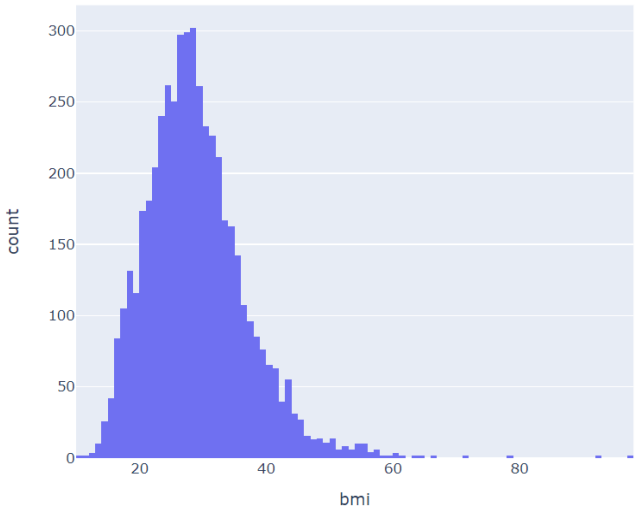
Research question인 stroke의 예측 및 질병 분류 문제 등의 해결을 위해, 전처리 과정의 일환으로 결측치 처리는 필수적이다. 특히, 문제 해결 방법으로서 머신러닝 모델링을 활용할 경우, 결측치를

잘못된 숫자로 대체하거나 아예 제거할 경우, 해당 변수로 인해 모델이 데이터를 제대로 학습하지 못할 가능성이 크다. 따라서, 해당 데이터의 분포를 확인하거나, 시계열 데이터라면 앞,뒤 레코드의 평균값을 활용한다거나 하는 다양한 방법을 고려하여 결측치 처리를 진행해야한다.

전체 dataframe의 결측치를 다음의 방법으로 확인해본 결과 bmi컬럼에서만 총 201건 결측치를 발견하였다. 총 5110건의 레코드 중 201의 결측치는 4%에 해당하는 숫자로 제거하기 보다는 특정 통계량으로 대체하여 데이터의 수를 유지하는 것이 옳은 방향이라고 판단하였다. 어떤 통계량으로 결측치를 대체할지 결정하기 위해 bmi의 히스토그램을 참고하였다.

```
[ ]  df.isnull().sum()

gender                0
age                   0
hypertension          0
heart_disease         0
ever_married          0
work_type             0
Residence_type        0
avg_glucose_level     0
bmi                 201
smoking_status        0
stroke                0
dtype: int64
```



EDA를 통해 확인한 바와 같이 대체로 정규분포를 따른다고 볼 수 있겠으나 70이상의 매우 큰 bmi의 레코드들을 몇 건 확인할 수 있었다. 이에 이상치에 덜 민감한 median으로 결측치를 채워주었다.

```
[ ]  med = np.median(np.array(df['bmi'].dropna()))
     med

     28.1

[ ]  df = df.fillna(med)
```

```
[ ]  df.isnull().sum()

gender                0
age                   0
hypertension          0
heart_disease         0
ever_married          0
work_type             0
Residence_type        0
avg_glucose_level     0
bmi                   0
smoking_status        0
stroke                0
dtype: int64
```

3.4 Data Sampling

분류 모델의 데이터 학습이 적절하게 이루어지도록 하려면, 분류 대상이 되는 target label의 수를 비슷하게 맞추어야한다. 분류 라벨이 한쪽으로 치우쳐져 있는경우, 예를 들어 99%의 0과 1%의 1인 라벨의 데이터의 경우 모든 데이터를 0으로 분류만 해도 99%의 정확도를 가지게 되고, 이는 1% 라벨 1을 분류하기 위한 모델의 목적을 달성했다고 보기 어렵다. 우리가 사용한 데이터셋은 의료 데이터로, 질환이 있는 환자가 일반인보다 적기 때문에 클래스의 불균형을 고질적인 문제로 가지고

있다. 데이터셋의 클래스의 불균형을 해소하여 모델의 목적을 달성하기 위해 Data Sampling 과정이 필요하다.

타겟 변수인 'stroke' 컬럼의 value를 count하였을 때, 1인 범주의 데이터가 너무 적기 때문에 undersampling 시 너무 많은 데이터를 제거하게 되어 정보 손실이 발생할 것을 우려하여 oversampling을 택하였다. 동일한 데이터를 단순히 증식시켜버리면 과적합이 발생하기 때문에, 적은 데이터 세트에 있는 개별 데이터들의 K 최근접 이웃(K Nearest Neighbor)을 찾아 이 데이터와 K개 이웃들의 차이를 일정 값으로 만들어 기존 데이터와 약간만 차이가 나는 새로운 데이터를 생성하는 SMOTE 기법을 사용하고자 하였다. SMOTE 기법은 연속형 변수에만 작동하지만, 우리 조에서 선택한 데이터의 경우, 범주형 변수를 포함하고 있었기 때문에 범주형 변수와 연속형 변수 모두에 적용할 수 있는 SMOTE-NC 기법을 사용하게 되었다.

```
oversample = SMOTENC([0, 4, 5, 6, 9], random_state = 2022)
x_over, y_over = oversample.fit_resample(x, y)
print(y_over.value_counts())
```

```
1    4861
0    4861
Name: stroke, dtype: int64
```

범주형 변수를 가지는 컬럼의 인덱스 배열을 전달하여 data sampling 모델을 정의하고 이를 x와 y에 대해 학습시켜 x_over, y_over을 반환받는다. data sampling을 거친 타겟 변수 y_over의 분포를 살펴 보았을 때, 1인 범주의 데이터와 0인 범주의 데이터의 개수가 같아진 것을 확인할 수 있다.

3.5 Data Partitioning

모델이 데이터를 학습하는 것 뿐 아니라, 실제 데이터에 모델이 잘 예측하는지 확인이 필요하다. 이를 위해 train data와 test data를 분리하여 train data에서만 모델의 학습을 진행하고, 실제 데이터에 대한 예측 성능을 test data로 확인한다. 머신러닝 모델링에서는 추가로 validation data를 분리하기도 하는데, train data를 모델이 학습하고, 모델의 파라미터 튜닝을 위해 validation data를 활용하여 최적의 파라미터 search를 진행하고, 최종적인 test data로 최종 예측 성능을 확인한다. 본 실습에서 활용한 데이터 셋의 경우 data sampling 이전 총 레코드가 5110건으로 많지 않은편에 속하기 때문에 현재 단계에서는 validation set을 분리하지 않았다. 이러한 경우 validation은 차후 모델링을 할때에 k-fold cross validation을 활용하여 진행할 수 있다.
Data sampling 이후 row 수가 9722개로, 유의미한 수의 test set을 확보하기 위하여, train:test=7:3의 비율로 데이터를 나누었다. 또한 타겟 변수인 'stroke'는 범주형 변수이기에 stratified sampling을 사용할 수 있는데, random sampling과 달리 데이터 비율을 반영할 수 있어 stratified sampling을 사용하였다.

```
[35] x_train, x_test, y_train, y_test = train_test_split(x_over, y_over, train_size=0.7, test_size=0.3, stratify = y_over, random_state = 2022)
```

```
[37] print('train set: %d, test set: %d' %(len(y_train), len(y_test)))

     train set: 6805, test set: 2917
```

```
print('train set 1 비율: %f, 0 비율: %f' %(y_train.value_counts(1)[0], y_train.value_counts(1)[1]))
print('test set 1 비율: %f, 0 비율: %f' %(y_test.value_counts(1)[0], y_test.value_counts(1)[1]))

train set 1 비율: 0.500073, 0 비율: 0.499927
test set 1 비율: 0.500171, 0 비율: 0.499829
```

train set의 데이터 개수가 6805개, test set의 데이터 개수가 2917개로 7:3 비율로 data partition이 된 것을 확인할 수 있다. 또한, train set과 test set의 데이터 비율이 소수 셋째짜리까지 같아 거의 동일한 비율을 가진 것을 확인할 수 있다.

3.6 Outlier detection

3.6.1 noise 제거

EDA 과정에서 'gender'라는 컬럼의 전체 5110개의 데이터 중, 2994건이 여성, 2115건이 남성, 1건이 'Other'인 것을 확인하였다. 이는 전체에서 비율이 매우적고, 그 의미를 분석하기에 부적절한 값이다. 따라서 노이즈로 판단하고 해당 값을 제거해주었다.

3.6.2 outlier

EDA 과정의 boxplot에서 적지 않은 수의 outlier을 확인할 수 있었다. 이에 정확한 이상치의 개수를 탐지하고자 함수를 작성했고, outlier의 index가 총 331개로 많은 것을 알 수 있었다. 따라서 outlier를 제거하지 않았다. 그러나 수치형 변수들의 특성별로 스케일이 다르기 때문에 올바른 분석이 어려우므로 이를 조정하는 과정이 필요하다.

```
[63]  1 for i in list(numeric_feature):
      2   outlier_index = detect_outlier_test(x_train, i)
      3
      4 outlier_index
```

```
Int64Index([ 708,  191, 6386,  163,  263, 2081, 8505, 8050,  113, 2016,
             ...
            2630, 1798, 2136, 3959, 1621, 9591, 7622, 7417, 4456, 6777],
           dtype='int64', length=331)
```

3.7 Scaling

특성별로 데이터의 스케일이 다르다면 올바른 분석이 어렵다. 따라서 모든 특성의 범위(또는 분포)를 같게 만들어주는 Scaling 과정이 필요하다. 수치형 변수인 'age', 'avg_glucose_level', 'bmi'에 대해서 Scaling을 진행하였다.

3.7.1 Standard Scaling

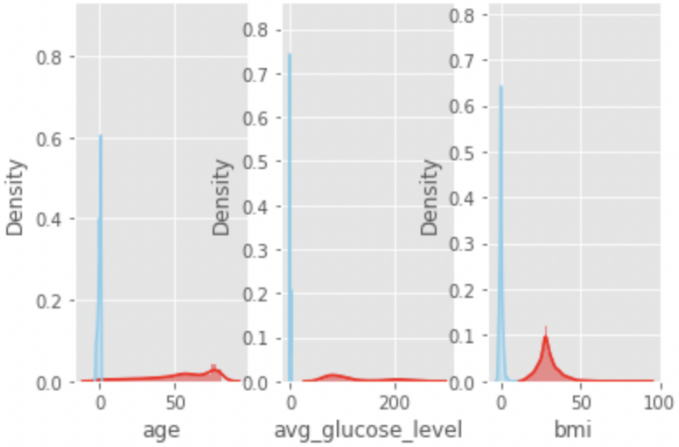
Standard Scaler는 각 특성의 평균을 0, 분산을 1로 변경하여 특성의 스케일을 조정하는 방법이다

.

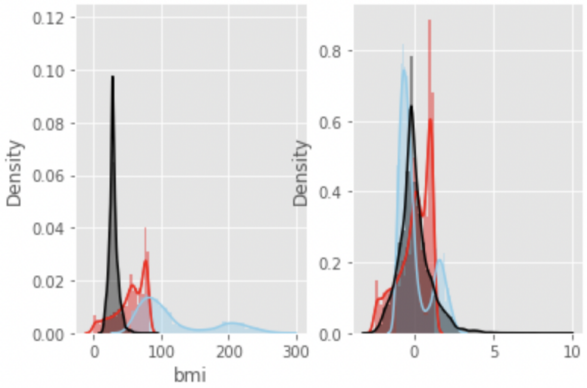
$$StandardScaler() = \frac{x - \mu}{\sigma}$$

$(x : 입력\ 데이터, \mu : 평균, \sigma : 표준편차)$

데이터에서 평균을 빼고 표준편차로 나누는 방식을 사용한다. 그러나 이상치가 있을 경우 평균과 표준편차가 영향을 받기 때문에 균형잡힌 척도를 보장할 수 없어 이상치에 민감하다. 이에 Scaling 전후를 시각화하여 확인해보았다.



위의 plot은 각 수치형 변수들의 스케일링 전,후 분포를 한 그래프에 나타낸 것이다. 왼쪽부터 'age', 'avg_glucose_level', 'bmi'의 그래프이다. red 그래프는 스케일링 전 분포이고 skyblue 그래프는 스케일링 후 분포이다.



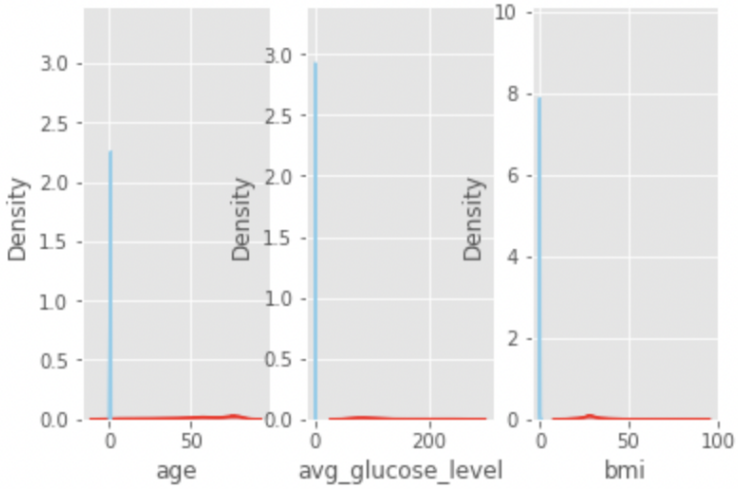
왼쪽 plot은 수치형 변수들의 스케일링 전의 분포이고 오른쪽 plot은 수치형 변수들의 스케일링 후이다. 각 그래프는 색깔별로 red = age, skyblue = avg_glucose_level, black = bmi을 나타낸다.

3.7.2 MinMax Scaling

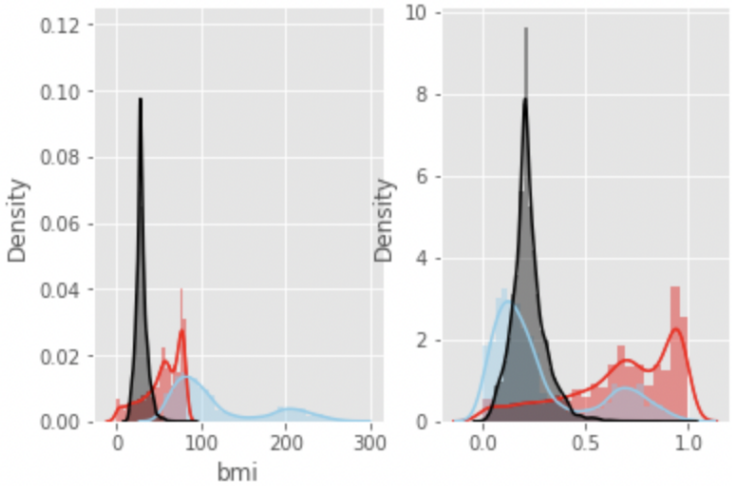
$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

MinMax Scaler는모든 feature의 값이 0과 1사이에 존재하도록 특성의 스케일을 조정하는 방법이다. 각 데이터에서 min값을 빼주고 max와 min의 차이로 나누어주는 방법이다. 그러나 out

lier가 존재하는 경우 Min, Max 값이 영향을 받기 때문에 이상치에 민감하다. 이에 Scaling 전후를 시각화하여 확인해보았다.



위의 plot은 각 수치형 변수들의 스케일링 전,후 분포를 한 그래프에 나타낸 것이다. 왼쪽부터 'age', 'avg_glucose_level', 'bmi'의 그래프이다. red 그래프는 스케일링 전 분포이고 skyblue 그래프는 스케일링 후 분포이다.



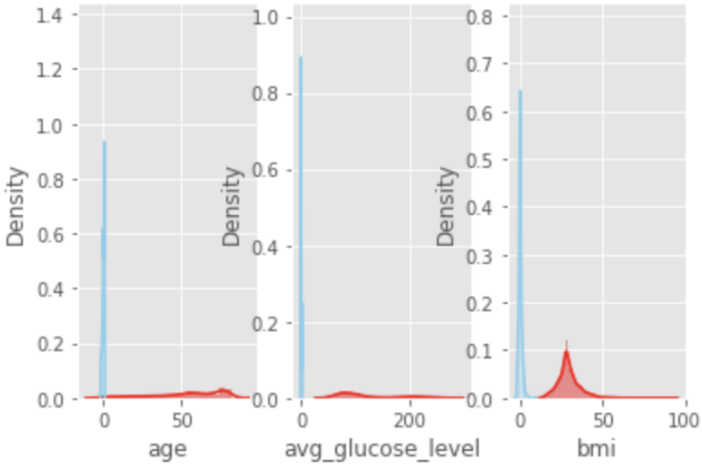
왼쪽 plot은 수치형 변수들의 스케일링 전의 분포이고 오른쪽 plot은 수치형 변수들의 스케일링 후이다. 각 그래프는 색깔별로 red = age, skyblue = avg_glucose_level, black = bmi을 나타낸다. 스케일링 후 feature별 크기가 다른 Scaler에 비해 유사하므로 현재 데이터셋에 사용하기 적합하지 않다.

3.7.3 Robust Scaling

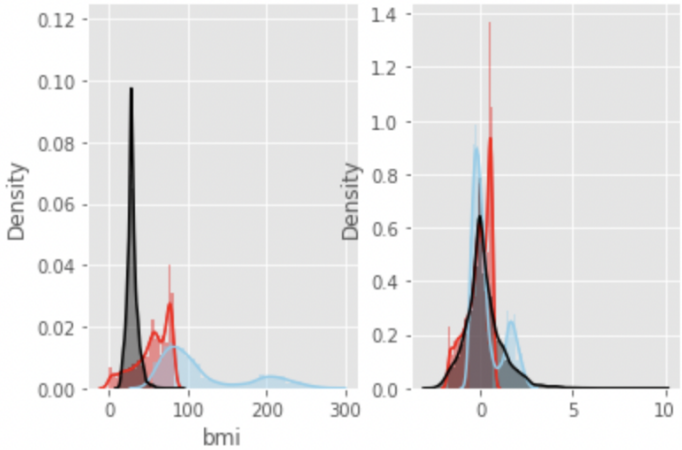
$$X_{\text{scale}} = \frac{x_i - x_{\text{med}}}{x_{75} - x_{25}}$$

Robust Scaler는 중앙값과 사분위수를 사용하여 아웃라이어의 영향을 최소화하는 스케일링 방법이다. 각 데이터에서 중앙값을 빼주고 IQR로 나누는 방법을 사용한다. 현재 데이터 셋에서는

age, bmi등의 컬럼에서 이상치가 존재하기때문에 이상치에 민감한 robust Scaler를 사용하여 중앙값이 0, 최대1, 최소 -1인 분포로 변환하였다.



위의 plot은 각 수치형 변수들의 스케일링 전,후 분포를 한 그래프에 나타낸 것이다. 왼쪽부터 'age', 'avg_glucose_level', 'bmi'의 그래프이다. red 그래프는 스케일링 전 분포이고 skyblue 그래프는 스케일링 후 분포이다.



왼쪽 plot은 수치형 변수들의 스케일링 전의 분포이고 오른쪽 plot은 수치형 변수들의 스케일링 후이다. 각그래프는 색깔별로 red = age, skyblue = avg_glucose_level, black = bmi을 나타낸다.

Standard Scaler, MinMax Scaler, Robust Scaler 3가지 Scaler를 사용하여 Scaling을 진행해주었다. MinMaxScaler의 경우 스케일링 후 feature별 크기가 다른 Scaler에 비해 유사하므로 사용하기 적합하지 않았다. Standard Scaler와 Robust Scaler의 경우 분포에 큰 차이가 존재하지는 않지만, 현재 수치형 데이터의 경우 이상치가 많이 존재하기 때문에 이상치에 덜 민감한 Robust Scaler를 사용하였다.

3.8 Transformation

수치형 변수의 왜도와 첨도를 살펴본 결과 다음과 같았다.

	skew	kurtosis
age	-0.780358	-0.349065
bmi	1.055340	3.362659
stroke	4.193284	15.589736

왜도와 첨도를 조정하는 공식적인 기준은 정해진 바가 없으나, 주로 왜도가 -2에서 2, 첨도가 -7에서 7의 범위 내에 있을 때 조정이 필요 없다고 본다.[3] 세 변수의 왜도와 첨도가 조정이 필요 없는 범위 내에 있기 때문에 데이터 변환은 진행하지 않았다.

3.9 Categorical encoding

모델 학습시 string data는 숫자형으로의 변환이 필요하다. 모델은 자연어 형식의 데이터를 학습할 수 없기 때문이다.

전체 데이터 중 'gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status' 5개의 변수는 문자열 값을 가진 범주형 변수였기 때문에 이에 대한 인코딩 과정이 필요했다.

	gender	ever_married	work_type	Residence_type	smoking_status
0	Male	Yes	Private	Urban	formerly smoked
1	Female	Yes	Self-employed	Rural	never smoked
2	Male	Yes	Private	Rural	never smoked
3	Female	Yes	Private	Urban	smokes
4	Female	Yes	Self-employed	Rural	never smoked

그 중 2개의 값을 가지는 gender, ever_married, Residence_type 변수는 scikit-learn의 LabelEncoder() 함수로 변환하였다. 다음은 변환 후의 라벨 설명이다.

Gender	
Male	Female
1	0
Residence_type	
Rural	Urban
0	1

[3] https://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/Simon

Ever married	
No	Yes
0	1

Residence_type	
Rural	Urban
0	1

나머지 변수인 work_type과 smoking_status는 각각 "children", "Govt_job", "Never_worked", "Private", "Self-employed", 그리고 "formerly smoked", "never smoked", "smokes", "Unknown" 등 2개 이상의 값을 가지기 때문에 get_dummies 변수로 One-hot 인코딩을 실행하였다.

4. Conclusion

4.1 RQ1

“흡연자의 경우, 비흡연자에 비해 뇌졸중에 걸릴 가능성이 상대적으로 높은가?”

본 실습에서 활용한 데이터의 경우 smoking status라는 컬럼으로 환자의 흡연여부가 기록되어 있었는데, 총 5110개의 데이터 중 1554건이 Unknown으로 기록되어있어 이 데이터 셋을 그대로 적용하여 해당가설을 검정하기에는 무리가 있다. 모델링을 위해서는 해당 데이터 컬럼을 활용하며 데이터셋의 크기는 유지하고자 get_dummies 변수로 one-hot encoding 하였지만, 해당 가설을 위해서는 차후 smoking_status==”Unknown” 인 레코드를 drop한 후 나머지 레코드들에 대해 흡연자 중 뇌졸중 환자의 비율, 흡연여부와 뇌졸중 여부의 상관관계 등을 확인 및 검정할 수 있다.

4.2 RQ2

“기저질환이 있을 경우 뇌졸중에 걸릴 가능성이 높은가?”

EDA과정에서의 범주형 변수간의 상관관계를 확인하는 cramers’v를 통해 개별 기저질환과 뇌졸중 간의 상관관계 계수를 구해볼 수 있었다. 하지만 이는 한가지 기저질환과 뇌졸중, 즉, 2개 변수 사이의 상관관계로 여러 기저질환의 복합적 작용이 뇌졸중을 일으키는 지에 대한 상관관계는 확인할 수

없었다. 앞서 진행한 전처리를 바탕으로 분류모델 생성과 예측 결과 분석을 통해 위 research question에 대한 해답을 제시할 수 있다.

4.3 RQ3

“성별, 나이, 각종 질병 여부, 거주 지역이나 결혼 여부 등의 생활양식을 통해 뇌졸중의 예측이 가능한가?”

위 Research question은 여러가지 변수를 활용한 분류문제로 볼 수 있다. decision tree나 random forest classifier, svm등의 머신러닝 알고리즘을 활용한 문제 해결을 생각해볼 수 있는데, 이러한 머신러닝 알고리즘을 활용하기 위해서는 데이터의 전처리 과정이 매우 중요하다. 데이터 전처리를 어떻게 진행하는지에 따라 모델이 데이터를 편향적으로 학습하기도 하고, 노이즈가 제거되지 않아 학습이 잘 이루어지지 않기도 한다. 앞서 수행한 전처리 과정을 바탕으로 차후 예측모델을 생성하여 그 성능 확인과 변수 중요도 확인을 통해 위 research question에 대한 해답을 제시할 수 있다.

5. Further Study

머신러닝 기술이 여러 분야로 확산되면서 의료 데이터에도 머신러닝을 도입하는 추세가 강해졌다. 뿐만 아니라 코로나 19의 대유행 이후 병원 방문에 많은 제약이 생기면서 집에서도 의료 서비스를 제공받는 원격 진료에 대한 수요가 증가하게 되었고 많은 가능성이 있는 분야로 각광받게 되었다. 직접 병원을 방문하지 않고도 병을 진단하고 예방할 수 있다면 의사와 환자 모두에게 편리하고 효과적인 의료 서비스를 제공할 수 있을 것이다.

본문에서는 비록 전처리 이후 모델링 단계는 진행하지 않았으나, 로지스틱 회귀나 SVM 등의 분류 모델을 적용하여 과연 변수별로 뇌졸중과 유의미한 관계가 있는지 파악함은 물론, 뇌졸중 여부를 예측할 수 있을 것이다. 클래스가 불균형한 데이터를 오버샘플링했을 때 우려되는 과적합 문제가 이 연구에서도 발생하는지, 만약 과적합되었다면 어느 정도로 되었는지까지 실험해볼 수 있을 것이다.

각 변수의 유의성을 검증하기 위해 통계적 검정을 진행할 수도 있을 것이다. ‘검정'이란 데이터로 부터 주어지는 정보를 이용하여 모수에 대해 추측한 가설이 통계적으로 유의한지 알아보는 것이다. 데이터에 따라 다양한 분포를 사용하여 검정할 수 있다. 현재 Research Question에서는 각 변수가 유의한지 판단하는 것이 목표이다. 따라서 본 데이터 셋을 활용할 경우에는 귀무가설을 ‘흡연여부와 뇌졸중 발병여부는 독립이다.’로 설정하고 대립가설을 ‘흡연여부와 뇌졸중의 발병여부는 독립이 아니다.’로 설정하여 검정을 진행할 것이다. 유의수준에 따른 검정결과를 살펴보았을 때 p-value가 0.05보다 작다면 귀무가설을 기각하고, 각 변수는 독립이 아니라는 결과를 얻게 된다. 만약 설명 변수들 간의 독립여부를 조사했을 때 독립이 아니라면 다중공선성이 존재한다는 것도 알 수 있을 것이다. 이에 따라 추가적인 가설검정을 진행하거나 다중공선성을 해결하는 방법을 진행할 수 있을 것이다.

더 나아가, 연구의 향후 발전 가능성까지 기대해볼 수 있다. 현재 다룬 데이터는 환자의 기본 정보만을 담고 있지만, 이외에도 휴대용 임상 기기로 혈압, 혈당 등의 데이터를 실시간으로 전달받아 분석한다면 더 정밀한 예측이 가능할 것이다. 또한, 뇌졸중 발병 환자의 CT, MRI 영상을 수집하여 이미지 처리 딥러닝 기술을 적용한다면 뇌졸중의 전조증상을 탐지하고 조기에 예방할 수 있을 것으로 기대한다.