

개인데이터분석프로젝트

뇌졸중 영향 변수 심층 분석 및 모델링

Deep Analysis and Modeling of Stroke Impact Factors

한성대학교 빅데이터트랙

임다솔

Email : lastromeo51@naver.com

Github : <https://github.com/Dasollm>

목차

- 
- 1 분석 개요와 목표
 - 2 Dataset 설명
 - 3 분석 과정 – Preprocessing & Modeling with Logistic Regression, SVM, and KNN
 - 4 결과 해석
 - 5 방안 제시
 - 6 참고 문헌

뇌졸중

뇌기능의 부분적 또는 전체적으로 급속히 발생한 장애가
상당 기간 이상 지속되는 것으로,
뇌혈관의 병 이외에는 다른 원인을 찾을 수 없는 상태

“ **뇌졸중(stroke)은 19세 이상의 성인에게서 나타나는 5대 사망원인의 질환 중 하나이며(통계청, 2021),
건강보험심사평가원은 최근 5년간 뇌혈관질환 진료현황 결과 를 보면, 뇌졸중 환자 수는 18년 591,946명 대비 22년
634,177명으로 연평균 1.7% 증가하는 추세이다(건강보험심사평가원, 2023).
그리고 인구의 노령화로 인해 매년 증가하고 있으며(Li, et al, 2021), ... ”

**최형욱. "가상현실훈련이 뇌졸중 환자의 균형, 보행 및 뇌파에 미치는 영향." 국내박사학위논문 남부대학교 일반대학원, 2024. 광주

허혈성 뇌졸중 환자 중 3.5시간 내 병원을 방문한 사람은 26.2%에 불과했으며,
골든타임 내 방문과 직결되는 재개통치료(정맥내 혈전용해술·동맥내 혈전제거술) 비율은 전체 환자 중 16.3%에 그쳤다.
또 병원 도착 시간이 늦어질수록 재개통치료를 받는 환자가 급격하게 줄어들었다.
4.5시간 내 병원에 도착한 뇌졸중 환자의 42% 정도가 재개통치료를 받은 반면,
4.5시간 이후 방문한 환자는 치료받는 비율이 10.7%로 급감했다.

출처 : 의협신문

뇌졸중 골든타임 방문 불과 26%
뇌졸중 환자의 퇴원 시 사망률은 2.6%로 다른 OECD 국가의 치명률과 비교했을 때 최상위권

국민대상, 국가적 차원의 뇌졸중 예방 방안과 대책의 필요성이 매우 크다.

목표

- ① 변수의 중요도 분석: 각 변수가 뇌졸중 발병에 미치는 영향을 분석한다. 변수의 중요도를 통해 어떤 변수가 가장 중요한지 식별한다.
- ② 모델링 통한 결과 도출: 변수 중요도를 제공하며, 변수의 영향을 해석하는 데 유리한 Logistic Regression Model을 이용한다.
- ③ 결과 해석 및 시각화: 모델의 결과를 해석하고, 중요한 변수 및 그 영향을 시각화하여 직관적으로 이해할 수 있도록 한다.
- ④ 예방 및 관리 전략 제안: 분석 결과를 바탕으로 뇌졸중 발병을 예방하고 관리하기 위한 전략을 제안한다.

변수 설명

★ id : 고유id

♀ gender : 성별 (남자:0, 여자:1, 기타:2)

😊 age : 나이

AB hypertension : 고혈압 여부

❤ heart_disease : 심장병 여부

👤 ever_married : 결혼 상태 (미혼:0, 결혼한 적 있음:1)

💻 work_type : 근무 형태 (공무원:0, 무직:1, 직장인:2, 자영업자:3)

🏠 residence_type : 주거 형태 (농촌:0, 도시:1)

💉 avg_glucose_level : 평균 혈당치

🍔 bmi : 체질량 지수

🚬 smoking_status : 흡연 여부 (과거 흡연자:0, 비흡연자:1, 현재 흡연자:2, 알 수 없음:3)

🧠 stroke : 뇌졸중 경험 여부

기타 변수 설명

- 답변자가 성별을 밝히지 않았길 원했을 경우 성별 부분 기타로 처리되었음.
- Age에서 18세 이하는 뇌졸중 영향이 특히 낮은 연령대로 이후 데이터셋에서 삭제되어 데이터 분석에서 제외될 것임.

데이터셋 일부 예시

```
df = pd.read_csv('/content/healthcare-dataset-stroke-data.csv')  
df.head(3)
```



	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1

1) 타겟 변수 설정

Stroke, 뇌졸중 경험 여부는 데이터 분석에서 타겟 변수로서 이용되어야 했기 때문에 따로 체크해주었다. 뇌졸중 전력이 없는 답변자는 약 95%, 뇌졸중 전력이 있는 답변자는 약 5%인 것으로 확인하였다.

```
[ ] df['stroke'].dtype
```

```
⇒ dtype('int64')
```

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

```
⇒ 0
```

```
[ ] df['stroke'].value_counts(dropna=False)
```

```
⇒ stroke
0    4861
1     249
Name: count, dtype: int64
```

```
[ ] df['stroke'].value_counts(dropna=False, normalize=True)
```

```
⇒ stroke
0    0.951272
1    0.048728
Name: proportion, dtype: float64
```


2) 기타 변수 데이터 처리

age, avg_glucose_level, bmi와 같은 구간변수를 따로 빼주었다. 간단한 데이터 확인을 진행하였다.

```
cols = ['age', 'avg_glucose_level', 'bmi']
df[cols].dtypes
```

```
age          float64
avg_glucose_level  float64
bmi          float64
dtype: object
```

```
[ ] pd.options.display.float_format = '{:.2f}'.format
df[cols].describe()
```

```

count  5110.00      5110.00  4909.00
mean    43.23       106.15   28.89
std     22.61       45.28    7.85
min      0.08       55.12   10.30
25%     25.00       77.25   23.50
50%     45.00       91.88   28.10
75%     61.00      114.09   33.10
max     82.00      271.74   97.60
```

```
df['age'] > 18
```

```

0      True
1      True
2      True
3      True
4      True
...
5105   True
5106   True
5107   True
5108   True
5109   True
Name: age, Length: 5110, dtype: bool
```

```
[ ] c = df['age'] > 18
df[c].head(3)
```

```

   id  gender  age  hypertension  heart_disease
0  9046   Male  67.00            0              1
1  51676  Female  61.00            0              0
2  31112   Male  80.00            0              1
```

2) 기타 변수 데이터 처리

범주형 변수만 따로 저장하여 결측값 확인하였으나 없었으므로, 따로 처리하지 않았다.
미성년자는 데이터 분석에서 제외할 것으로, 18세 초과 데이터만 따로 저장하였다.
그 결과 5110줄에서 데이터가 4194줄로 줄었다.
변수명 통일 위해 Residence_type을 residence_type으로 변경하였다.



```
# 범주형 변수들만 cols1에 저장.
```

```
cols1 = ['gender', 'hypertension', 'heart_disease', 'ever_married', 'work_type', 'residence_type', 'smoking_status']  
df1[cols1].isnull().sum()
```



```
gender          0  
hypertension    0  
heart_disease   0  
ever_married    0  
work_type       0  
residence_type  0  
smoking_status  0  
dtype: int64
```

```
[ ] len(df[c])
```

```
↻ 4194
```

```
[ ] df1 = df[c]  
df1.shape
```

```
↻ (4194, 12)
```

```
[ ] df1 = df1.rename(columns={'Residence_type' : 'residence_type'})
```

3) 결측값 50% 이상 변수 제거

결측값이 있는 column으로 bmi가 도출되었지만, 50% 넘지 않는 관계로 제거하지 않았다.

```
df1.isna().any()[lambda x: x]
```

```
bmi    True  
dtype: bool
```

```
[ ] df['bmi'].isnull().mean()
```

```
0.03933463796477495
```

4) 요약 통계 및 도수분포표 점검

왜도와 첨도 확인 결과, 정상적인 범주 내에 데이터가 있음을 확인하였다.



#정규분포 확인

#-3에서 3 사이일 시 정상분포로 봄

```
df1[cols].skew()
```



```
age          0.03
avg_glucose_level  1.45
bmi          1.24
dtype: float64
```



#첨도 확인

#-10에서 10 사이일 시 정상으로 봄

```
df1[cols].kurtosis()
```



```
age          -1.02
avg_glucose_level  1.09
bmi          3.45
dtype: float64
```

5) 이상값 제거

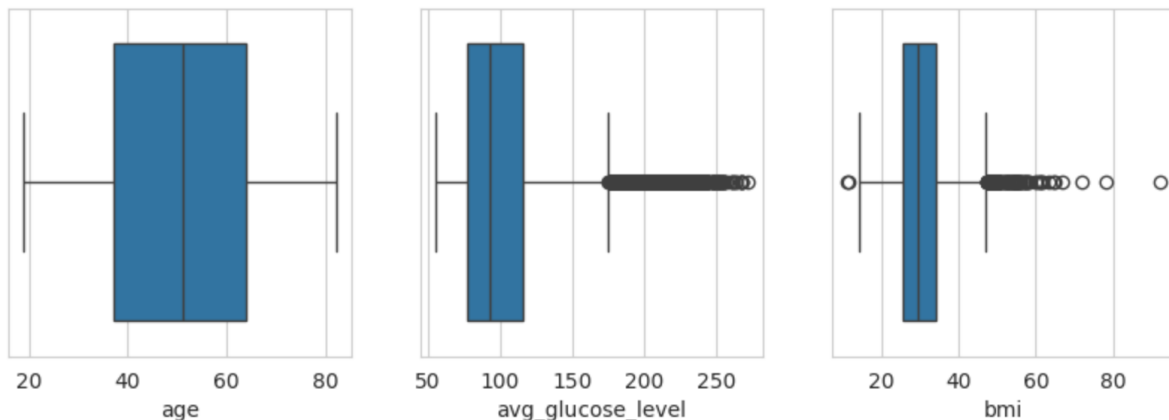
구간변수 age, avg_glucose_level, bmi 이상값을 제거하기 전, 그래프를 통해 대략적으로 변수 분포를 확인하였다.

Boxplot을 통해 대략적으로 avg_glucose_level, bmi에 이상값이 있음을 확인하였다.

```
sns.set_style('whitegrid')
```

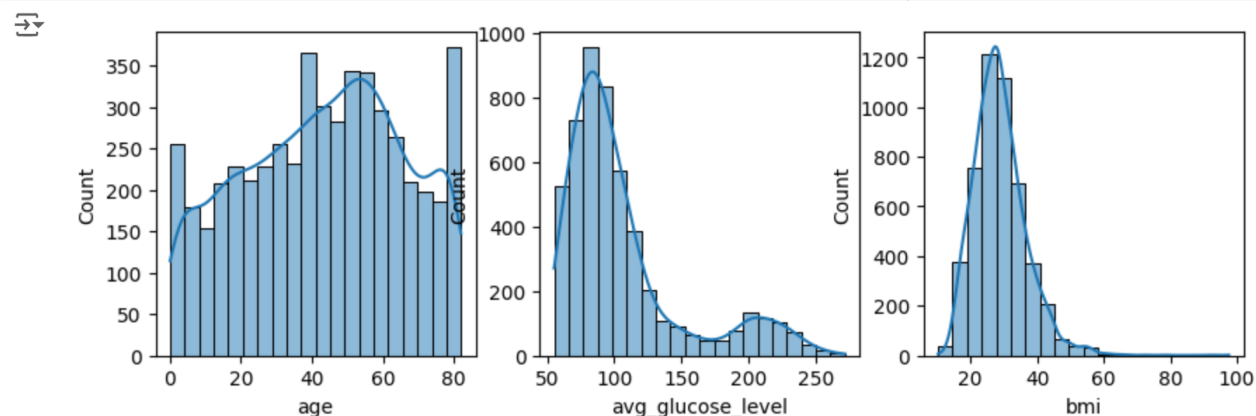
```
fig, axes = plt.subplots(1, 3, figsize=(10, 3))
sns.boxplot(ax=axes[0], x='age', data=df1)
sns.boxplot(ax=axes[1], x='avg_glucose_level', data=df1)
sns.boxplot(ax=axes[2], x='bmi', data=df1)
```

<Axes: xlabel='bmi'>



```
import matplotlib.pyplot as plt
import seaborn as sns

fig, axes = plt.subplots(1, 3, figsize=(10, 3))
sns.histplot(ax=axes[0], data=df, x="age", kde=True, bins=20);
sns.histplot(ax=axes[1], data=df, x="avg_glucose_level", kde=True, bins=20);
sns.histplot(ax=axes[2], data=df, x="bmi", kde=True, bins=20)
plt.show()
```



5) 이상값 제거

사분위값 이용, 이상값을 제거한 후 df2에 새롭게 저장하였다.

```
Q1 = df1[['age', 'avg_glucose_level', 'bmi']].quantile(0.25)
Q3 = df1[['age', 'avg_glucose_level', 'bmi']].quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
age          27.00
avg_glucose_level  38.80
bmi           8.70
dtype: float64
```

```
[ ] Lower = Q1-3.0*IQR
Upper = Q3+3.0*IQR
print(Lower)
```

```
age          -44.00
avg_glucose_level -38.99
bmi          -0.60
dtype: float64
```

```
[ ] print(Upper)
```

```
age          145.00
avg_glucose_level 232.64
bmi           60.30
dtype: float64
```

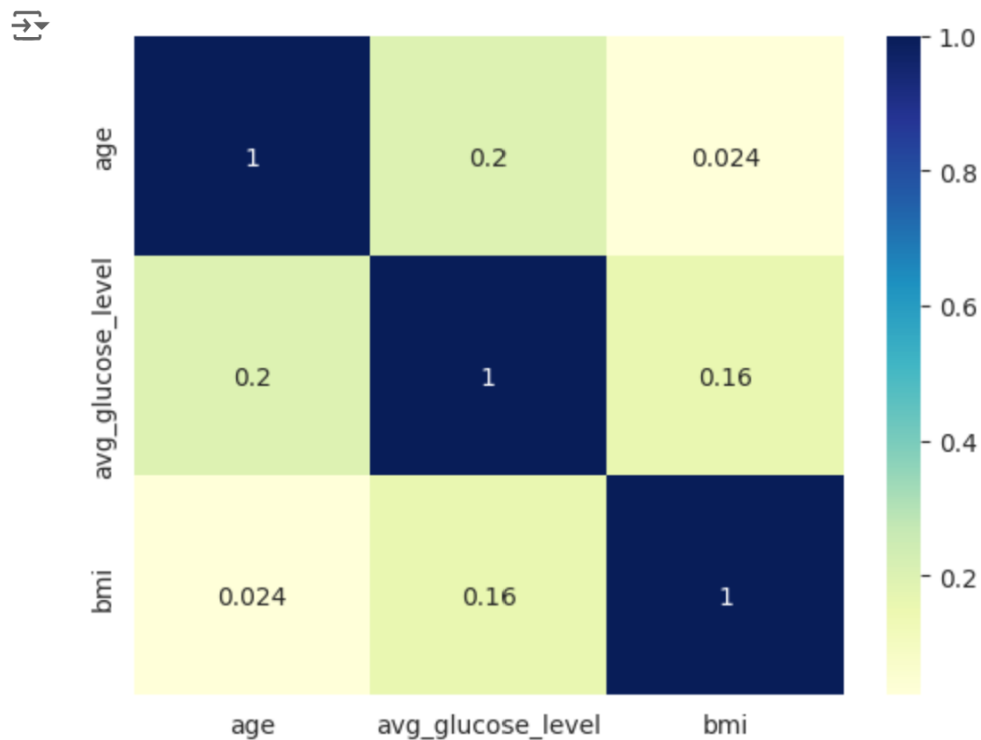
```
[ ] c1 = df1['avg_glucose_level'] <= 232.64
c2 = df1['bmi'] <= 60.3
df2 = df1[c1 & c2]
df2.shape
```

```
(3915, 12)
```

6) 상관관계 검토

상관관계 검토하였을 때 크게 문제되는 데이터가 없음을 확인하였다.

```
[ ] import seaborn as sns  
    corr = df2[cols].corr()  
    annot_kws = {"ha": 'center', "va": 'top'}  
    sns.heatmap(data=corr, annot=True, annot_kws=annot_kws, cmap="YlGnBu")
```



7) Ordinal Encoder 이용한 추가 변수 처리

입력변수로 수치형 데이터 이용하기 위해, Ordinal Encoder 통해 변수 데이터를 이진변수로 처리하였다.
이후 encoding 전 column은 제거하였다.

```
---- ever_married ---
ever_married
Yes    0.79
No     0.21
Name: proportion, dtype: float64
```

```
---- work_type ---
work_type
Private      0.66
Self-employed 0.19
Govt_job     0.16
Never_worked 0.00
Name: proportion, dtype: float64
```

```
---- residence_type ---
residence_type
Urban    0.51
Rural    0.49
Name: proportion, dtype: float64
```

```
---- smoking_status ---
smoking_status
never smoked    0.42
formerly smoked 0.20
Unknown         0.20
smokes          0.18
Name: proportion, dtype: float64
```

```
---- gender ---
gender
Female  0.61
Male    0.39
Other   0.00
Name: proportion, dtype: float64
```

```
from sklearn.preprocessing import OrdinalEncoder
df3['ever_married_encoded'] = OrdinalEncoder().fit_transform(df3['ever_married'].values.reshape(-1,1))
df3.groupby(['ever_married', 'ever_married_encoded']).size()
```

```
ever_married  ever_married_encoded
No            0.00                804
Yes           1.00               3111
dtype: int64
```

```
df3['gender_encoded'] = OrdinalEncoder().fit_transform(df3['gender'].values.reshape(-1,1))
df3.groupby(['gender', 'gender_encoded']).size()
```

```
gender  gender_encoded
Female  0.00           2402
Male    1.00           1512
Other   2.00             1
dtype: int64
```

```
df3['work_type_encoded'] = OrdinalEncoder().fit_transform(df3['work_type'].values.reshape(-1,1))
df3.groupby(['work_type', 'work_type_encoded']).size()
```

```
work_type  work_type_encoded
Govt_job   0.00                611
Never_worked 1.00                 2
Private    2.00               2566
Self-employed 3.00                736
dtype: int64
```

```
list(df3.columns)
```

```
['age',
 'hypertension',
 'heart_disease',
 'avg_glucose_level',
 'bmi',
 'stroke',
 'ever_married_encoded',
 'gender_encoded',
 'work_type_encoded',
 'residence_type_encoded',
 'smoking_status_encoded']
```


8) 전처리 완료된 csv 도출

전처리 완료한 csv를 따로 저장하였다.

```
df3.to_csv('/content/stroke-dataset-preprocessed.csv', index=False)
```

1) 다중범주형변수 변환

전처리된 csv 데이터셋을 모델링을 위해 불러왔다.

다중범주형변수를 독립적인 이진형 더미 변수로 이용하기 위해 원핫인코딩을 진행하였다.

```
[ ] import pandas as pd
import numpy as np
df_lr = pd.read_csv('/content/stroke-dataset-preprocessed.csv')
df_lr.head(3)
```



	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke	ever_married_encoded
0	67.0	0	1	228.69	36.6	1	
1	80.0	0	1	105.92	32.5	1	
2	49.0	0	0	171.23	34.4	1	

다음 단계: [df_lr 변수로 코드 생성](#)

☒ [추천 차트 보기](#)

```
[ ] cols = ['gender_encoded', 'work_type_encoded', 'smoking_status_encoded']
```

```
[ ] df_lr2 = pd.get_dummies(df_lr, columns=cols)
```



df_lr2.info()



<class 'pandas.core.frame.DataFrame'>

RangeIndex: 3915 entries, 0 to 3914

Data columns (total 19 columns):

#	Column	Non-Null Count	Dtype
0	age	3915 non-null	float64
1	hypertension	3915 non-null	int64
2	heart_disease	3915 non-null	int64
3	avg_glucose_level	3915 non-null	float64
4	bmi	3915 non-null	float64
5	stroke	3915 non-null	int64
6	ever_married_encoded	3915 non-null	float64
7	residence_type_encoded	3915 non-null	float64
8	gender_encoded_0.0	3915 non-null	bool
9	gender_encoded_1.0	3915 non-null	bool
10	gender_encoded_2.0	3915 non-null	bool
11	work_type_encoded_0.0	3915 non-null	bool
12	work_type_encoded_1.0	3915 non-null	bool
13	work_type_encoded_2.0	3915 non-null	bool
14	work_type_encoded_3.0	3915 non-null	bool
15	smoking_status_encoded_0.0	3915 non-null	bool
16	smoking_status_encoded_1.0	3915 non-null	bool
17	smoking_status_encoded_2.0	3915 non-null	bool
18	smoking_status_encoded_3.0	3915 non-null	bool

dtypes: bool(11), float64(5), int64(3)

memory usage: 286.9 KB

2) 기준더미변수 처리

각각 '남자', '무직', '흡연한 적 없음'을 기준 더미변수로 정하였다.

```
df_lr2.to_csv('/content/stroke-dataset-preprocessed-dummy.csv', index=False)
```

```
[ ] df_lr_dummy = pd.read_csv('/content/stroke-dataset-preprocessed-dummy.csv')  
df_lr_dummy.shape
```

```
⇒ (3915, 16)
```

```
[ ] df_lr_dummy.dtypes
```

```
⇒ age                float64  
hypertension          int64  
heart_disease         int64  
avg_glucose_level     float64  
bmi                   float64  
stroke                int64  
ever_married_encoded  float64  
residence_type_encoded float64  
gender_encoded_1.0    bool  
gender_encoded_2.0    bool  
work_type_encoded_0.0  bool  
work_type_encoded_2.0  bool  
work_type_encoded_3.0  bool  
smoking_status_encoded_0.0 bool  
smoking_status_encoded_1.0 bool  
smoking_status_encoded_3.0 bool  
dtype: object
```

3) Under Sampling과 데이터셋 분할

타겟변수와 나머지 변수들로 나눈 후, Random Under Sampling 진행하였다.
데이터셋을 학습용과 테스트용으로 50%로 분할하였다.

```
[ ] data = df_lr_dummy.drop(['stroke'], axis=1)
    target = df_lr_dummy['stroke']

    from imblearn.under_sampling import RandomUnderSampler
    undersample = RandomUnderSampler(sampling_strategy=0.333, random_state=51)
    data_under, target_under = undersample.fit_resample(data, target)

[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(
        data_under, target_under, test_size=0.5, random_state=329, stratify=target_under)

    print("X_train shape:", X_train.shape)
    print("X_test shape:", X_test.shape)
```



```
X_train shape: (386, 15)
X_test shape: (386, 15)
```

4) 로지스틱 회귀 모델링

로지스틱 회귀 모델로 모델링을 진행한다.

Training set score는 약 78%, Test set score는 약 80%로 overfitting 문제가 발생하지 않은 듯 보인다.

```
[ ] from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score

    lr = LogisticRegression(solver='lbfgs', penalty='none', random_state=0, n_jobs=-1)
    model = lr.fit(X_train, y_train)
    pred = model.predict(X_test)

    print ("Logreg Training set score:{:.5f}".format(model.score(X_train, y_train)))
    print ("Logreg Test set score:{:.5f}".format(accuracy_score(y_test, pred)))
```



```
Logreg Training set score:0.77979
Logreg Test set score:0.79534
```

5) 로지스틱 회귀 모델링 & 그리드서치

로지스틱 회귀 모델 최적화를 위해 5-fold cross-validation으로 그리드서치를 진행하였다.
교차검증 정확도는 약 0.76%, 최적의 하이퍼파라미터 조합은 `penalty=None`, `solver=lbfgs`로 도출되었다.
`grid_lr.best_estimator_`를 통해 그리드서치로 찾은 최적의 모델을 반환했고,
이를 이용해 테스트 데이터셋 정확도를 80%로 얻을 수 있었다.
정확도가 높고 일관적이므로, 모델이 데이터를 잘 예측하고 있다는 타당한 지표로 받아들일 수 있게 되었다.
이 결과를 통해 최종적인 결론을 도출할 수 있을 것이다.

```
[ ] lr = LogisticRegression(solver='lbfgs', penalty='none', random_state=0, n_jobs=-1)

# Grid Search
from sklearn.model_selection import GridSearchCV
params = {'solver': ['lbfgs', 'saga'], 'penalty': ['none']}

grid_lr = GridSearchCV(lr, param_grid=params, scoring='accuracy', cv=5, n_jobs=-1,)
grid_lr.fit(X_train, y_train)

print("GridSearchCV max accuracy:{:.5f}".format(grid_lr.best_score_))
print("GridSearchCV best parameter:", (grid_lr.best_params_))
```

```
➡ GridSearchCV max accuracy:0.76157
GridSearchCV best parameter: {'penalty': 'none', 'solver': 'lbfgs'}
```

```
[ ] best_clf = grid_lr.best_estimator_
pred = best_clf.predict(X_test)
print("Accuracy on test set:{:.5f}".format(accuracy_score(y_test, pred)))
```

```
➡ Accuracy on test set:0.79534
```

6) Odds Ratio 도출

결론에 쓰일 Odds Ratio 표를 도출하였다. 이를 통해 결과 해석을 진행할 수 있게 되었으므로, 메인 모델링을 마무리한다.

```
feature_names = list(data.columns)
df_lr_ = pd.DataFrame(np.round(np.exp(best_clf.coef_),3).transpose(), index=feature_names, columns=['Odds_ratio'])
df_lr_coef = df_lr_.sort_values(by='Odds_ratio', ascending=False)
df_lr_coef
```



Odds_ratio



hypertension	2.239
heart_disease	1.886
smoking_status_encoded_3.0	1.186
age	1.073
smoking_status_encoded_1.0	1.053
residence_type_encoded	1.018
bmi	1.013
avg_glucose_level	1.003
gender_encoded_2.0	1.000
gender_encoded_1.0	0.830
ever_married_encoded	0.712
smoking_status_encoded_0.0	0.619
work_type_encoded_2.0	0.313
work_type_encoded_0.0	0.209
work_type_encoded_3.0	0.173

모델이 데이터셋을 잘 예측하는지 추가적으로 확인하기 위해 SVM 모델 Grid Search 진행

정확도가 높고 일관적이어서 모델이 데이터셋을 잘 예측하고 있다고 판단한다.

```
[ ] clf_mlp = MLPClassifier(max_iter = 2000, random_state = 0)

from sklearn.model_selection import GridSearchCV
params = {'solver':['sgd', 'lbfgs', 'adam'],
          'alpha':[0.0001, 0.001, 0.01, 0.1, 1],
          'activation':['tanh','relu', 'logistic']}

grid_mlp = GridSearchCV(clf_mlp, param_grid=params, scoring='accuracy', cv=5, n_jobs=-1)
grid_mlp.fit(X_train, y_train)

print("GridSearchCV max accuracy:{:.5f}".format(grid_mlp.best_score_))
print("GridSearchCV best parameter:", (grid_mlp.best_params_))
```

⇒ GridSearchCV max accuracy:0.76687
GridSearchCV best parameter: {'activation': 'relu', 'alpha': 0.0001, 'solver': 'sgd'}

```
[ ] best_clf = grid_mlp.best_estimator_
pred = best_clf.predict(X_test)
print("Accuracy on test set:{:.5f}".format(accuracy_score(y_test, pred)))
```

⇒ Accuracy on test set:0.76425

모델이 데이터셋을 잘 예측하는지 추가적으로 확인하기 위해 KNN 모델 Grid Search 진행

SVM과 동일하게 정확도가 높고 일관적이어서 모델이 데이터셋을 잘 예측하고 있다고 판단한다.

```
[ ] clf_knn = KNeighborsClassifier(n_neighbors=3)

from sklearn.model_selection import GridSearchCV
params = {'n_neighbors': range(3, 31)}

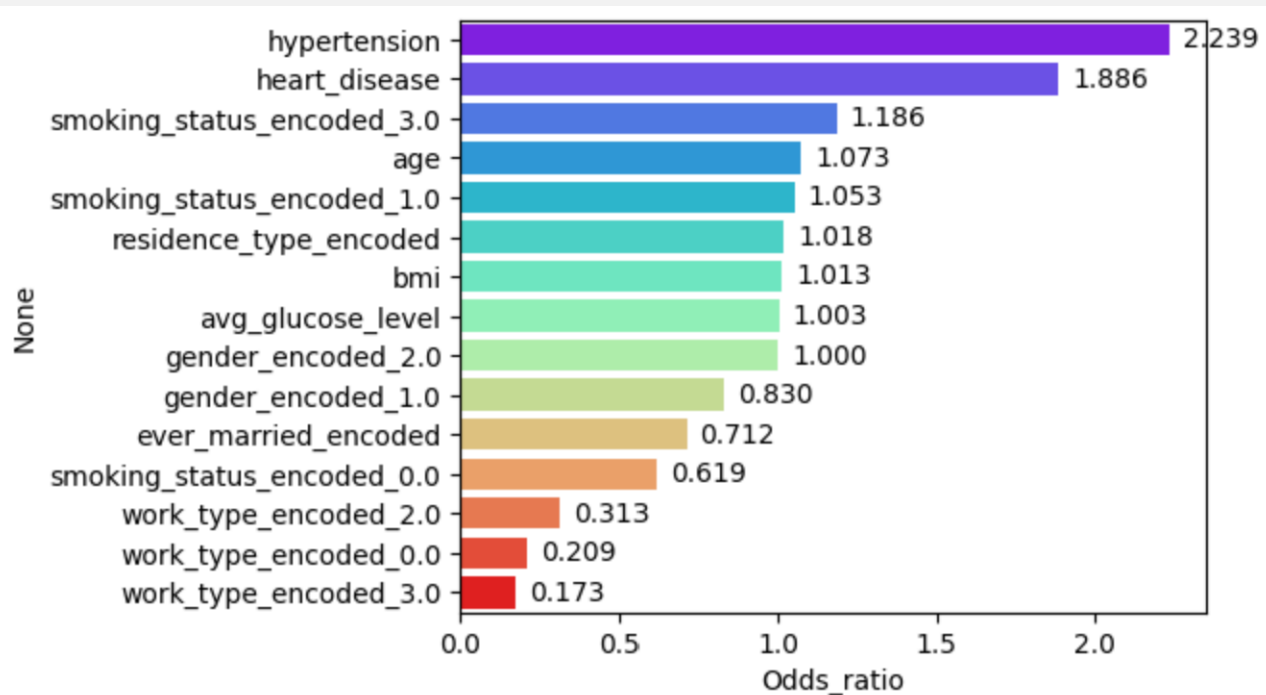
grid_knn = GridSearchCV(clf_knn, param_grid=params, scoring='accuracy', cv=3, n_jobs=-1)
grid_knn.fit(X_train, y_train)

print("GridSearchCV max accuracy:{:.5f}".format(grid_knn.best_score_))
print("GridSearchCV best parameter:", (grid_knn.best_params_))
```

⇒ GridSearchCV max accuracy:0.76690
GridSearchCV best parameter: {'n_neighbors': 18}

```
[ ] best_clf = grid_knn.best_estimator_
pred = best_clf.predict(X_test)
print("Accuracy on test set:{:.5f}".format(accuracy_score(y_test, pred)))
```

⇒ Accuracy on test set:0.76425



♀ gender : 성별 (남자:0, 여자:1, 기타:2)

😊 age : 나이

AB hypertension : 고혈압 여부

❤ heart_disease : 심장병 여부

👤 ever_married : 결혼 상태 (미혼:0, 결혼한 적 있음:1)

💻 work_type : 근무 형태 (공무원:0, 무직:1, 직장인:2, 자영업자:3)

🏠 residence_type : 주거 형태 (농촌:0, 도시:1)

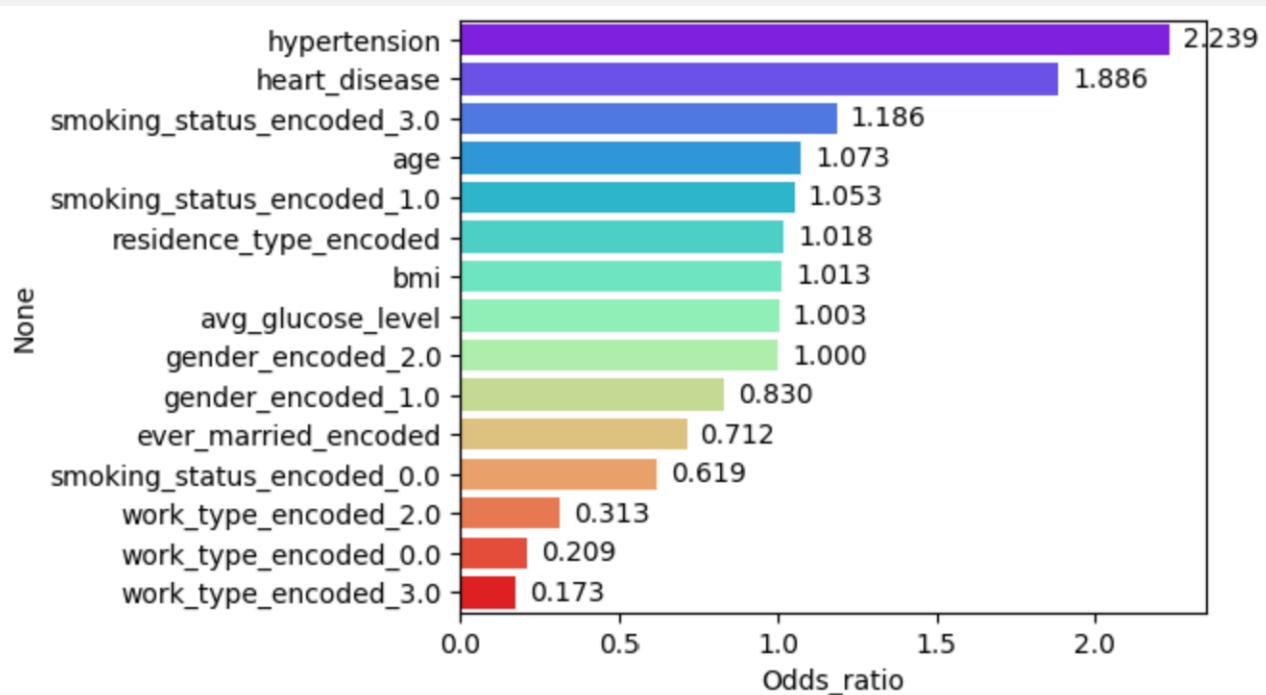
🩺 avg_glucose_level : 평균 혈당치

🍔 bmi : 체질량 지수

🚬 smoking_status : 흡연 여부 (과거 흡연자:0, 비흡연자:1, 현재 흡연자:2, 알 수 없음:3)

뇌졸중에 영향이 높은 변수 TOP6

1. 고혈압이 있는 경우 뇌졸중 발생 확률이 고혈압이 없는 경우보다 약 2.239배 더 높다.
2. 심장병이 있는 경우 뇌졸중 발생 확률이 심장병이 없는 경우보다 약 1.886배 더 높다.
3. 흡연 상태가 '알 수 없음'인 경우 뇌졸중 발생 확률이 다른 흡연 상태에 비해 약 1.186배 더 높다.
4. 나이가 1살 증가할 때마다 뇌졸중 발생 확률이 약 1.073배 증가한다.
5. 비흡연자의 뇌졸중 발생 확률이 다른 흡연 상태에 비해 약 1.053배 더 높다.
6. 도시에 거주하는 경우 뇌졸중 발생 확률이 농촌에 거주하는 경우보다 약 1.018배 더 높습니다.



♀ gender : 성별 (남자:0, 여자:1, 기타:2)

😊 age : 나이

AB hypertension : 고혈압 여부

❤ heart_disease : 심장병 여부

👤 ever_married : 결혼 상태 (미혼:0, 결혼한 적 있음:1)

💻 work_type : 근무 형태 (공무원:0, 무직:1, 직장인:2, 자영업자:3)

🏠 residence_type : 주거 형태 (농촌:0, 도시:1)

🩺 avg_glucose_level : 평균 혈당치

🍔 bmi : 체질량 지수

🚬 smoking_status : 흡연 여부 (과거 흡연자:0, 비흡연자:1, 현재 흡연자:2, 알 수 없음:3)

뇌졸중에 영향이 낮은 변수 TOP6

1. 자영업자의 뇌졸중 발생 확률이 무직인 경우보다 약 0.173배 낮다.
2. 공무원의 뇌졸중 발생 확률이 무직인 경우보다 약 0.209배 낮다.
3. 직장인의 뇌졸중 발생 확률이 무직인 경우보다 약 0.313배 낮다.
4. 과거 흡연자의 뇌졸중 발생 확률이 다른 흡연 상태에 비해 약 0.619배 낮다.
5. 결혼한 적이 있는 경우 뇌졸중 발생 확률이 미혼인 경우보다 약 0.712배 낮다.
6. 여자의 뇌졸중 발생 확률이 남자보다 약 0.830배 낮다.

고위험군 대상 뇌혈관 질환 검사 및 경동맥 초음파 검사 등의 활발한 권장과 지원

분석 결과
심장병과 고혈압 병력자에 대한
집중적인 지원이 필요할 것,

추가적으로
보험에 대해 잘 모르실 수 있는
노년층을 위해
지자체나 동사무소 차원의
실비 보험 정보 컨설팅 지원이
필요할 것

노인대상 보건정책의 확립과 연구 적극적 차원의 조사

** “조기 사망이나 또는 입원 중
거동 불편 등의 이유로
국민건강영양조사 대상에
포함되지 않았을 가능성도 염두에
...”

조사에서 소외되는 노년층 없도록
더욱 더 적극적인 사회적 차원의
조사가 필요할 것

사후처리가 아닌, 사전 예방을 지향,
조기 발견과 예방을 통한
노인 건강 지수 향상이 실현되도록
해야 할 것

실효성 있는 금연 프로그램의 지원과 홍보 교육의 필요성

과거 흡연자의 뇌졸중 발생 확률이
다른 흡연 상태에 비해 약 0.619배
낮은 분석 결과를 토대로
국가적 차원의 금연 홍보와 교육,
금연 프로그램의 정책적 지원이
현재보다 더 활발히 수반되어야 할 것

최형욱. "가상현실훈련이 뇌졸중 환자의 균형, 보행 및 뇌파에 미치는 영향." 국내박사학위논문 남부대학교 일반대학원, 2024. 광주

<https://www.doctorsnews.co.kr/news/articleView.html?idxno=154428> (의협신문)

문홍진. "노인의 주관적 건강수준에 영향을 미치는 요인." 국내박사학위논문 가천대학교 글로벌캠퍼스 일반대학원, 2024. 경기도

장진희. "성인 남성 뇌졸중 흡연자의 금연의도 관련요인." 국내석사학위논문 忠南大學校 大學院, 2020. 대전

